(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
6 June 2002 (06.06.2002)

PCT

(10) International Publication Number
WO 02/45315 A2

(51) International Patent Classification[7]: H04L

(21) International Application Number: PCT/US01/43140

(22) International Filing Date:
21 November 2001 (21.11.2001)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
09/724,025 28 November 2000 (28.11.2000) US

(71) Applicant: MICROMUSE INC. [US/US]; 139 Townsend Street, 5th Floor, San Francisko, CA 94107 (US).

(72) Inventors: HERRING, David; 45 Church Road East Crowthorne, Berkshire RE45 7ND (GB). CARROLL, John, D.; 4543 North O'Connor Road, Apt. 1249, Irving, Texas 75062 (US). O'GRADY, Rehan; Basement Flat, 32 Addison Avenue, Hollamd Park, London W11 4QR (GB). COLEMAN, Neil; 40 Belle Vue Road, Quary Bank, West Midlands DY5 1AD (GB). MARKS, Felix; 251 Mayall Road, London SE24 OPQ (GB).

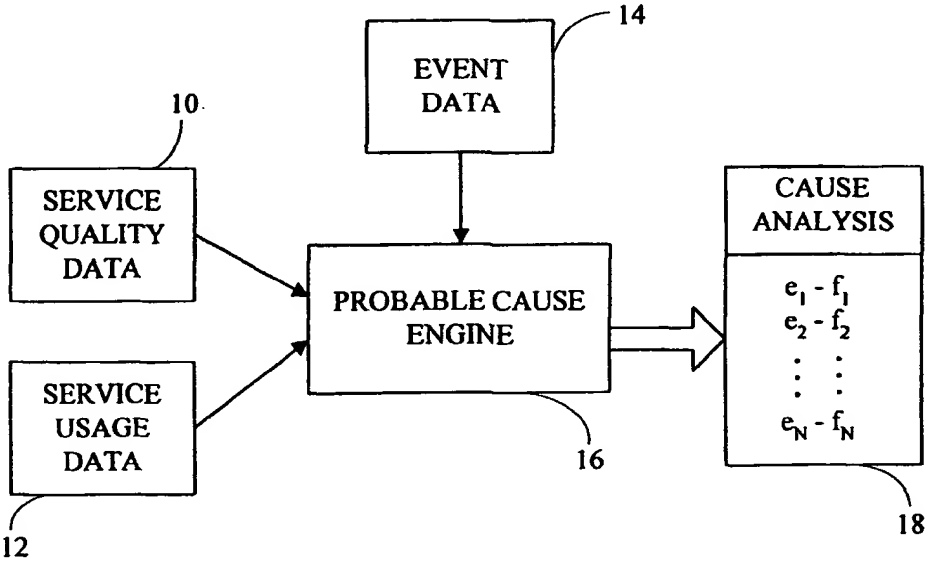
(74) Agent: OSTROW, Seth. H.; Brown Rausman Millstein, Felder & Steiner LLP, 900 Third Avenue, New York 10022 (US).

(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, OM, PH, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, UZ, VN, YU, ZA, ZW.

(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR,

[Continued on next page]

(54) Title: METHOD AND SYSTEM FOR PREDICTING CAUSES OF NETWORK SERVICE OUTAGES USING TIME DOMAIN CORRELATION



(57) Abstract: Method and system are described for predicting the likely causes of service outages using only time information, and for predicting and the likely costs of service outages. The likely causes are found by defining a narrow likely cause window around an outage based on service quality and/or service usage data, and correlating service events to the likely cause window in the time domain to find a probability distribution for the events. The likely costs are found by measuring usage loss and duration for a given point during anoutage and using cost component functions of the time and usage to extrapolate over the outage. These cause and cost predictions supply service administration with tools for making more informed decisions about allocation of resources in preventing and correcting service outages.

WO 02/45315 A2

GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:

— *without international search report and to be republished upon receipt of that report*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

METHOD AND SYSTEM FOR PREDICTING CAUSES OF NETWORK SERVICE OUTAGES USING TIME DOMAIN CORRELATION

COPYRIGHT NOTICE

BACKGROUND OF THE INVENTION

The invention disclosed herein relates to the field of fault event correlation and assessment in computerized services such as telecommunications networks or application programs. More particularly, the invention relates to methods, systems, and software for determining likely causes of service outages and assessing the costs of the service outages. The invention described herein is useful in correcting service outages, preventing future occurrences of similar service outages, and determining the appropriate level of resources which should be allocated to correct and prevent such outages.

Maintaining the proper operation of various types of computerized services is usually an important but difficult task. Service administrators are often called upon to react to a service failure by identifying the problem which caused the failure and then taking steps to correct the problem. To avoid wasting resources investigating the wrong problems, administrators must make accurate assessments as to the causes of failures. Because substantial time and resources are often required, administrators must also make accurate decisions as to when to allocate resources to the tasks of identifying problems and fixing them.

A number of tools are available to assist administrators in completing these tasks. One example is the NETCOOL® suite of applications available from Micromuse Inc., assignee of the present application, which allows network administrators to monitor activity on networks such as wired and wireless voice communication networks, intranets, wide areas networks, or the Internet. The NETCOOL suite logs and collects network events, including network occurrences such as alerts, alarms, or other faults, and then reports them to network administrators in graphical and text based formats. Administrators are thus able to observe network events on a real-time basis and respond to them more quickly. The NETCOOL suite

also includes network service monitors of various types which measure performance of a network so that, among other things, network resources can be shifted as needed to cover outages.

Even knowing that events are occurring in real time, however, administrators must still make judgments as to which events are responsible for causing service failures or outages and which service outages are worth the expenditure of resources to fix. Although experienced administrators can usually make reasonably accurate judgments, it is desirable to provide additional application tools which improve the chances that these judgments are accurate.

A number of existing systems attempt to correlate events with service failures. For example, U.S. Patent No. 5,872,911 to Berg describes a system that monitors a telecommunications network for faults and assesses fault data to determine the likely cause of the fault. The system accomplishes this by filtering and reducing the full set of fault data based on a correlation of various faults or alarms. The correlation is performed using a rules-based engine or knowledge base which determines or defines relationships among types of faults. The system then uses the filtered and reduced fault data to determine actual service impact on the network by determining whether a network outage occurred as a result of the fault. The system further determines which customers or equipment are affected by the network outage using conventional mechanisms which track network traffic.

As another example, U.S. Patent No. 5,748,098 to Grace describes a system which uses stored historical data concerning alarms and when they occur to determine the probability of a relationship between events occurring within a window of time. The window of time is either fixed or determined with reference to the nature of the network or the stored historical data. Additional patents, including U.S. Patent No. 5,646,864 to Whitney, U.S. Patent No. 5,661,668 to Yemini, and U.S. Patent No. 6,049,792 to Hart, describe still further schemes for attempting to correlate and relate network faults or alarms using expert systems, logic models, or complex causality matrices.

These patents, which are hereby incorporated by reference herein, describe systems which monitor and attempt to correlate faults in a network. However, among other things, these systems fail to take full advantage of available performance or usage data in correlating events. That is, the inventors have found that a more careful analysis of the level of usage of a service improves the correlation of events to service outages. There is therefore

a need for methods and systems for accounting for service usage information among other data to improve correlation between events and service failures.

Furthermore, there is a need for improved methods and systems for helping administrators make decisions about how to prioritize outages and allocate resources in the correction or prevention of service failures. Commonly assigned application serial no. 09/476,846 and U.S. Patent No. 5,872,911, discussed above, describe different systems for determining the impact of a service failure on customers or users. However, these systems do not quantify the impact in a way to provide the administrator with the ability to compare the effects of outages in different, unrelated services in order to prioritize the allocation of resources, or to perform a strict cost/benefit analysis for the allocation of the resources. Improved methods and systems are thus needed to quantify the cost of a service outage in such a way as to allow the cost to be compared to costs of other service outages in services or systems which may differ in type or use.

SUMMARY OF THE INVENTION

It is an object of the present invention to solve the problems described above with existing network analysis systems.

It is another object of the present invention to improve correlation between events and outages in computerized services.

It is another object of the present invention to improve the correlation without requiring substantive knowledge of the nature of the event and its effect on the service.

It is another object of the present invention to help service administrators make better decisions about how to apply resources to correct service failures.

It is another object of the present invention to quantify the overall cost of a service outage.

It is another object of the present invention to quantify the cost in a way which allows outages in different types of services or systems to be easily compared with one another.

It is another object of the present invention predict the cost of an outage during the outage.

Some of the above and other objects are achieved by a method for analyzing a potential cause of a service change, such as a service outage or service recovery. The method involves monitoring service quality of the service and usage amount of the service, and

determining a service change time window based at least in part upon a change in service quality between a first working state and a second, non-working state as well as upon a change in service usage amount. The service change time window encompasses at least part of a service outage, such as the onset or completion of the outage. As used herein, a service outage or failure is intended to include not only total failures and complete losses of service but also reductions in level of a service to a level considered to be unacceptable or requiring corrective action. The method further involves detecting an event and a time in which the event occurred, and computing a probability that the detected event caused the service change based at least in part on a relation between the event time and the service change time window.

The present invention provides a significant improvement over prior attempts to correlate faults or alarms in a number of ways. For example, in certain aspects of the present invention, three items of information are taken - service quality, service usage, and events - and they are correlated in the time domain, based only on their times. In addition, using both the service quality and service usage data to define a service change time window which is likely to contain or near the cause of a service outage the correlation between the detected event and the service outage.

The method is applicable to various types of computerized services, including: telecommunication networks such as voice or data telephony systems, wired or wireless telephony, email, or instant messaging; computer networks such as intranets, extranets, local or wide area networks, or the Internet; and application programs such as web server software, operating systems, financial system software, etc. The method may be implemented in a computer program which is stored on a computer readable medium such as a magnetic or optical disk and is executed to cause the computer to perform the method.

In some embodiments, the service quality is monitored by a service monitor which regularly polls the service to determine a binary classification of the service as either good or bad. Alternatively, the service quality monitor may identify a number of discrete states of the service, and the administrator selects which combination of the states represents a working state versus a non-working state. The service usage amount is measured by a usage meter, which may be part of the service monitor or another component, and which tracks the level of traffic, performance or usage of the service on a substantially continuous basis.

Since in these embodiments the service quality is monitored by polling, a finding of bad service at some point indicates that an outage occurred at some prior point in time after the last monitored level of good. In some embodiments, the service change time window is determined by first defining a service failure time window bounded by successive readings of good and bad service levels, and then narrowing the service failure time window based upon the service usage amount measured during that service failure time window. This combination provides a more accurate time window encompassing the outage and which more likely encompasses or correlates to the event causing the outage.

The probability that the detected event caused the service change may be computed in a number of ways. In some embodiments, one or more weighting functions are used to compute a probability distribution covering events occurring during the service change time window and within a given time period prior to the window. Although the probability distribution could include events occurring a relatively long time prior to the outage, the likelihood that these older events are responsible for the outage exponentially decreases with distance from the time window and at some point are too negligible to be considered. The weighting functions may be combined to produce a combined weighting function. Exemplary weighting functions used in various embodiments of the invention include:

- a time weighting function for each detected event which decreases exponentially with the distance between the detected event occurrence time or times and the service outage time window;
- a false occurrence weighting function for each event which decreases the probability of the event as the cause of the service outage for instances in which an event of the same type occurred outside the service change time window;
- a positive occurrence weighting function for each detected event which increases the probability of the detected event as the cause of the service change based on instances stored in a historical database in which a detected event of the same type occurred within a prior service change time window; and
- a historical weighting function for each detected event which increases the probability of the detected event as the cause of the service outage based

on instances in the historical database in which a detected event of the same type was identified as having caused a prior service change, perhaps by a user or some automated mechanism.

Some of the above and other objects are achieved by method for analyzing potential causes of a service change. The method involves determining a service change time window encompassing a change of service between a first working state and a service outage, the service change being determined at least in part based on measured service usage levels. For example, a service change may be detected by detecting a step change in service usage, without the need to resort to separate service quality data. A set of events occurring within a given time prior to and during the service change time window are detected, each occurrence of an event being associated with a time at which the event occurred. A probability distribution is computed for the set of events, which probability distribution determines for each event in the set the probability that the detected event caused the service change, the probability distribution being based at least in part on relations between the time of each event occurrence and the service change time window.

Some of the above and other objects of the invention are achieved by a network monitoring system having a service monitor for monitoring quality of service on the network, a usage meter for monitoring usage of the network, an event detector for detecting network events and times at which the network events occur, and a probable cause engine, coupled to receive data from the service monitor, usage meter, and event detector. The probable cause engine sets a service outage time window based upon data received from the service monitor and usage meter, the service outage time window encompassing an occurrence of a service outage in the network. The probable cause engine further determines which of the network events detected by the event detector is the most likely cause of the service outage based at least in part on the relation between the detected network event times and the service outage time window.

Some of the above and other objects of the invention are achieved by a method for quantifying the effect of a service outage over a first period of time. The method involves monitoring or measuring levels of service usage over time, defining a cost of outage time window. The cost of outage time window includes the first time period and a second time period following the first time period. The method further involves computing a cost of the outage as the difference between the measured service usage during the cost of outage time

window and service usage measured in a comparison window. The comparison window is substantially equal in time to the time of the cost of outage window and reflects a similar period of service activity as the cost of outage window, but has no service outage.

In some embodiments, the second period of time is the time in which the service usage returns or recovers from the outage to within a given percentage of a normal service usage level. This recovery should last for more than a mere instant to reflect a return to a state of relative equilibrium, as the service usage is likely to fluctuate following the restoration of service. For example, in an email system, a large number of messages may accumulate in the queue while the email system is down, and a this larger number of messages may be conveyed in a short period of time once the system is returned, resulting in very high volumes. In some embodiments, the second period of time is bounded by a maximum time, e.g., a multiple of the time of the outage.

In some embodiments, the cost of outage is computed in units of service usage. What constitutes a unit of usage depends upon the nature of the service. For example, if the service is a communication service conveying messages, such as telephony or email, the unit of usage is one or more messages. If the service is a network server providing data items such as web pages in response to requests such as from clients, the unit of usage may be a received request or data item provided by a server.

Units of usage may be normalized across different services by converting them to a value such as a monetary value, number of users, time value, or other value representing a loss resulting from not receiving the unit of usage. This may be accomplished by multiplying the units of usage by an assigned value of usage per unit. The normalized value of the service outage cost may then be used by administrators or an automated decision making program to compare costs of difference services and prioritize the allocation of resources across the services to prevent or repair service outages.

In some embodiments, the service usage is continuously monitored, including after the end of cost of outage window, and is used to measure the long term effects of the service outage. This is accomplished by computing the difference between the service usage following the cost of outage time window and a normal service usage level. If the service usage is substantially below or above the normal level following the outage, this is an indication that the outage is of the type that causes loss of use of the service, including loss of users.

Some of the above and other objects of the invention are achieved by a method for predicting a cost of an outage of a service such as during the outage itself. The method involves measuring the time duration for and service usage during the outage, and comparing the measured usage amounts to normal usage amounts measured under similar service conditions for a similar period of time where no service outage occurs, to thereby determine a usage loss amount. This usage loss amount is used to compute a predicted cost of the outage based at least upon a cost component, the cost component comprising a function of the measured time of the outage and measured usage loss amount. The predicted cost of service outage may be compared to a second predicted cost of outage value for a different service so that outages in the two services may be prioritized based on the compared costs.

In some embodiments, the cost components include a service demand cost component representing an affect on service usage based upon the duration of an outage, a customer retention cost component representing a number or percentage of customers lost due to the outage, an agreement penalty component representing penalties arising under one or more service agreements due to a service outage, and a trust loss component representing empirically collected assessments of cost due to loss of trust in the service. The service demand cost component may be computed by multiplying the measured usage loss by a usage loss curve which is a function of time duration of an outage and represents a predicted percentage usage due to an outage based on time duration of the outage. The usage loss curve may be generated using historical data derived from prior service outages.

The methods and systems of the present invention thus provide a service administrator with the ability to more accurately assess what events are likely to have caused a service outage, without the need to know very much about the nature of the events, and to make a more well informed prioritization of service outages and allocation of resources in correcting or preventing the events and corresponding outages.

BRIEF DESCRIPTION OF THE DRAWINGS

The invention is illustrated in the figures of the accompanying drawings which are meant to be exemplary and not limiting, in which like references are intended to refer to like or corresponding parts, and in which:

Fig. 1 is a data flow diagram showing a time-based event correlation methodology in accordance with one embodiment of the present invention;

Fig. 2 is a block diagram showing a service with service level and usage monitors in accordance with one embodiment of the present invention;

Fig. 3 is a block diagram showing a network event correlation system in accordance with one embodiment of the present invention;

Fig. 4 is a flow chart showing a process of correlating events with a service outage in accordance with one embodiment of the present invention;

Fig. 5 is a graphical representation of an exemplary set of service quality data;

Fig. 6 is a graphical representation of an exemplary set of service usage data;

Figs. 7-9 are graphical representations of techniques for defining a likely cause window using service level or service usage data in accordance with embodiments of the present invention;

Fig. 10 is a graphical representation of an exemplary time weighting function in accordance with one embodiment of the present invention;

Fig. 11 is a graphical representation of exemplary outages and events;

Fig. 12 is a table showing a probability distribution for the events shown in Fig. 11 to have caused one of the outages shown in Fig. 11;

Fig. 13 is a graphical representation of a technique for defining a likely cause window for a service recovery using service level or service usage data in accordance with one embodiment of the present invention;

Fig. 14 is a flow chart showing a process of predicting the cost of an outage in accordance with one embodiment of the present invention;

Figs. 15A-15B contain a flow chart showing a process of quantifying the cost of a completed outage in accordance with one embodiment of the present invention;

Figs. 16-19 are graphical representations of techniques for quantifying outage costs in accordance with one embodiment of the present invention;

Figs. 20-22 are graphical representations of exemplary cost of outage analyses performed in accordance with embodiments of the present invention;

Figs. 23-34 are graphical representations of techniques for predicting the cost of an outage during the outage in accordance with embodiments of the present invention;

Fig. 25 is an exemplary usage loss curve as a function of time for use in predicting the cost of an outage in accordance with one embodiment of the present invention; and

Figs. 26-28 are graphical representations of exemplary usage loss scenarios for use in generating a usage loss curve in accordance with one embodiments of the present invention.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

Embodiments of the present invention are now described with reference to the drawings in the Figures.

One goal of these embodiments is to discover the likely cause of a service outage or service recovery by identifying which event or events occurring in the service are most likely to have caused the outage or recovery. For the sake of simplicity, the following description will focus on predicting costs of service outages, but the techniques and system described herein can be applied to service recovery, as explained further below.

Referring to Fig. 1, three sources of data are used to achieve this goal - service level or quality data 10, service usage data 12, and events and timestamps associated with the events 14. A probable cause engine 16 receives these three sources of data and processes them using various heuristics as described herein to build a probability distribution 18 over the events in the system, based on which events are most likely to have caused the outage. In some embodiments, the probable cause engine uses the service level data 10 and service usage data 12 to define a time window, sometimes referred to herein as a Likely Cause Window (LCW), which has a high probability of containing the event which caused the outage. The events 14 are then related to the outage based upon a time domain correlation. By virtue of the probability distribution, the probable cause engine 16 orders the events and assigns a discrete probability to each event, so that a service administrator may more accurately assess which event caused the outage.

Event data 14 consists of a timestamp and identifier for each event. In some embodiments, the probabilistic analysis performed by the probable cause engine 16 relies only on the time and type of an event, and does not involve or require any further knowledge of what a particular event consists of or the nature of the event. For purposes of the description below, an event is often distinguished from an event type. An event generally refers to a single alarm, alert, or fault generated in or detected by the system monitoring the service. The same happening in the service may generate multiple events. An event type, on the other hand, generally refers to the network happening that generated the events. Events 14 having the same identifier all come under the umbrella of a single event type.

In addition, a specific terminology is used herein in which a set, E, is a set of events. Members of this set are referenced with double subscript notation. Several events may, for example, have the same identifier and hence all represent alarms generated by a single event type. An event type is referred to by a single subscript. Different events of a particular type are distinguished with the second subscript. For example, events e_{11} and e_{12} would refer to two events of type e_1. These events have the same identifier but different timestamps. Event e_{11} represents the first chronological event of type e_1 in the set E, event e_{12} represents the second chronological event of type e_1, event e_{21} represents the first chronological event of type e_2 in the set E, and so on.

The service level or quality data and service usage data are collected through any conventional devices depending upon the particular service. For example, referring to Fig. 2, in one embodiment of the invention, a service accessed by a number of end users 22 is monitored by service level monitors 24 as well as usage meter(s) 26. As shown, the usage meter or meters measure the total amount in which the service is used. This measurement will depend upon the nature of the service. For example, in a telecommunication service, the usage meter 26 measures the number of messages being conveyed by the service in an hour or other time period. Such a telecommunication service may include a wireless telephony system using the smart message service protocol, in which the usage meter measures the number of messages conveyed over the system. Similarly, in a computer network, the amount of traffic may be measured over a given period. For a service consisting of an application software program, the usage may be measured in any appropriate manner, such as time spent, number of memory calls over time, number of documents accessed over time, etc.

The service level monitors 24 monitor the status of the service on a polled basis. That is, they poll the service on a regular basis to determine whether the service is working. In some embodiments, this inquiry is binary in nature, resulting in a status of either working or non-working, sometimes referred to herein as a status of good or bad service. Alternatively, the service level monitors 24 may detect several types of service status, and an outage would be defined as desired as a transition from one status in which the service is considered to be working properly to a second status in which it is not providing an acceptable level of service. Multiple discrete service monitors 24 may be used to monitor the quality of service from different locations. For example, for a network accessible from several remotely located sites, each site would have a service monitor 24 to monitor the

quality of service being received at that site, while the traffic on the entire network would be measured by usage meters 26.

As explained further below, the service quality and service usage data are singly or jointly used to detect when an outage has occurred in the service. As used herein, an outage of a service is generally a period or window of time during which positive affirmation of the service being good or acceptable can not be obtained. When an outage is detected by a service monitor 24, for example, an outage time window contains one or more bad polls and is bounded at each end by two good polls. This guarantees that the outage window encompasses the entire outage incident.

One embodiment of the present invention implemented in a computerized network environment is shown in Fig. 3. In this embodiment various activity on a network 30 is monitored by a network monitoring system 32, such as the NETCOOL suite available from Micromuse Inc., as described above. Other commercially available network monitoring systems 32 may be used, as would be understood by one skilled in the art. The network monitoring system 32 works in conjunction with one or more service quality monitors 24, such as Internet Service Monitors, a set of active monitors which work with the NETCOOL suite to provide availability information on Internet service elements including DNS, HTTP, LDAP, POP3, RADIUS, SMTP, and TCP Port Monitoring, or NT Service Monitors, which work with the network monitoring system 32 to collect management events from a network operations center regarding Windows NT environments.

Further, a set of service usage meters 26A, 26B, and 26C work with the network monitoring system 32 to track the amount of activity seen at different locations on the network 30. In particular, in some embodiments service specific usage meters 26A track activity on different services 34 available within the network, including, for example, a cache server, an email server, Radius, HTTP/HTML, LDAP, and other applications and services available to end users over the network 30. Other, network edge usage meters 26B measure activity in and out of the network, including, for example, a usage meter 26B detecting activity at a firewall monitoring application which collects security information, a usage meter 26B at Cisco Net Flow, a usage meter 26B at a PIX firewall, and others as known to those skilled in the art.

Furthermore, network sniffing usage meters 26C, sometimes referred to as snooping agents, capture and listen to packets on the network received at a network interface.

Examples of conventional network sniffing usage meters include Snoop, Etherreal, and Etherfind. Etherfind, for example, examines packets traversing a network interface and outputs a text file describing the traffic. In the file, the text describing a single packet contains values such as protocol type, length, source, and destination. Further packet filtering can be done on the basis of protocol, e.g., IP, ARP, RARP, ICMP, UDP, TCP, and filtering can also be done based on the source, destination addresses as well as TCP and UDP port numbers.

Events are detected by the network monitoring system 32 through probes or similar conventional mechanisms and are stored in an event object server 40. The object server is an object-oriented data repository where information on events is stored, viewed and managed. Events are stored as objects on the server 40 which can be more easily manipulated for processing. A gateway 42 allows a mediator program suite 44 to share the event data stored in the object server 40. The mediator program suite 44 includes the probable cause engine 16 as well as a cost of outage engine 46 for quantifying the effect of an outage, as described in greater detail below. An administrator desktop and user interface 48 allows a network administrator to interact with the mediator 44 and execute the probable cause engine 16 and cost of outage engine 46.

The operation of one embodiment of the probable cause engine 16 is now described with reference to the flow chart in Fig. 4 and the graphical diagrams in Fig. 5-13. Referring to Fig. 4, the probable cause engine receives service quality data from service quality monitors and service usage data from the service usage meters, step 60. This received data is used to detect the occurrence of an outage, step 62. When an outage is detected, the probable cause engine defines a time window, sometimes referred to herein as the likely cause window or LCW, using the service quality data, service usage data, or preferably both, step 64. The probable cause engine then retrieves event data from the event object server for events which occurred during the defined likely cause window and for a period prior to the LCW, step 66. The events retrieved are each assumed to have a non-zero probability of having caused the outage, and it is further assumed that at least one of the retrieved events caused the outage. The engine then computes a probability distribution for the events which maps the probabilities that each event caused the detected outage, step 68. These probabilities are then associated with the events and output to the administrator in a suitable form.

Particular implementations of steps in this methodology are now described with reference to the graphs shown in Figs. 5-12. Figs. 5 and 6 illustrate sample data sets for service quality and service usage, respectively. As shown in Fig. 5, service quality data consists of a series of points 80 regularly spaced at points in time. As shown, the service quality is monitored at one of two levels or states - good and bad - although a number of quality states might be detected. The classification of a quality state as good or bad is defined in the service monitor configuration by its discrete measurement classification. For purposes of simplicity in this application, it is understood that a transition from one of such states to another of such states is considered to represent a failure or outage in the service, with the definition of what loss of service constitutes an outage being a design consideration.

Fig. 6 shows service usage data consisting of a greater number of points measured on a significantly more continuous basis. The service usage data points may be measured on an entirely continuous basis, e.g., every email message conveyed or every HTTP request is counted, or sufficient samplings of usage may be taken to accurately represent the actual usage of the service.

As explained above, the service quality and service usage data are used to detect outages and define the likely cause window. For purposes of this discussion, time windows are understood to generally cover continuous subsets of the time line. In the notation used below, square brackets denote inclusion of endpoints in the subset, while round brackets mean the end point is not included. For example, given two times a and b, a window (a,b] denotes a time window starting but not including time a and ending at and including time b. In addition, for an event e, and a window, W, the notation $e \in W$ indicates that the timestamp associated with event e falls within the window W.

Figs. 7 and 8 illustrate two ways in which outages are detected and the likely cause window is defined. One method of detecting an outage is through the service quality data received from a service monitor. As illustrated in Fig. 7, when a poll of the service monitor registers as bad for the first time following one or more pollings of good quality, the probable cause engine interprets that the service went bad at some stage between the current bad poll and the previous good poll. The time window between these two polls is defined as the service failure window or SFW. In the diagram in Fig. 7, the SFW is the set (b,c]. The time period between two readings of good in the service quality data in which one or more bad quality readings are monitored is defined as the outage window. In Fig. 7, the outage

window is the set (b,e). As described in greater detail below, the outage window is used among other things by the cost of outage engine to predict the cost of the outage and to quantify the cost of the outage in terms which may be compared to other service outages for purposes of prioritizing the application of resources in response to the outages.

In the absence of further information, the probable cause engine defines the LCW as the SFW plus a small, preset time tolerance consisting of the windows [a,b] and [c,d]. The LCW thus represents [a,d] in the diagram in Fig. 7. Alternatively, the time tolerances are set to zero, in which case the LCW = the SFW.

Alternatively, the probable cause engine detects an outage using only the service usage data, without necessarily resorting to the service quality data. In this embodiment, illustrated in Fig. 8, the engine detects a usage step change and verifies based on the step change whether the service has gone bad. To detect a usage step change, the probable cause engine detects a changepoint in a univariate time series, the time series consisting of the discrete measurements of units of usage. If a change in the service quality has already been detected by a service monitor, the SFW is defined as explained above as the period during which the service outage started. The usage step change is found in this window using an offline changepoint detection algorithm. Some such algorithms are described in Booth, N.B. and Smith A.F.M. *A Bayesian approach to Retrospective Identification of Change Points*, J. Econ., 19, (1982), which article is hereby incorporated herein by reference. If service quality data is not available, the probable cause engine analyzes the service usage data itself to detect an outage. This analysis employs an online changepoint detection algorithm. Some online changepoint detection algorithms are described in Basseville, M and Nikiforov, I.V., *Detection of abrupt Changes, Theory and Applications*, Prentice Hall, Englewood, New Jersey (1993), which article is also incorporated by reference herein.

As with an outage detection based on service quality change alone, the LCW may be defined by the probable cause engine as the time window from briefly before the usage step change up to and including the step change. However the outage is detected, clearly getting a tight bound on when the actual outage occurred is important to determining which event caused the outage. The smaller the LCW, the more likely it is that the probable cause engine will be able to pinpoint the event which caused the outage. Preferably, the

LCW should be the smallest time window that can reasonably be specified in which the data indicates the outage occurred.

In some embodiments, then, the probable cause engine uses the usage step change to provide a tighter bound on a SFW previously established using service quality data. As above, a service monitor detects that the service quality has gone bad. With the addition of usage data, however, the probable cause engine puts a much tighter bound on the SFW to produce a narrower LCW. This is illustrated in Fig. 8, in which the service quality data of Fig. 7 is shown in conjunction with usage data. It will be recognized by one skilled in the art that, for purposes of comparing the two types of data and graphing them together in Fig. 8 and later drawings, either or both of the service quality data and service usage data have been normalized. The probable cause engine analyzes the usage data for the time period within the SFW found through the service quality data and detects a usage data step change. The LCW is then defined as the period of the step change plus time tolerances.

As a result, in some embodiments the definition of the likely cause window is based on the service quality and service usage data received regarding a specific service. In other embodiments, the probable cause engine considers outages in other services to achieve a more refined definition of the service failure window and ultimately the likely cause window. This is based on a theory that the cause of an outage under consideration often causes outages in other, related services. Thus, referring to Fig. 9, the engine identifies related services as services experiencing an outage whose service failure window overlaps with the service failure window for the outage under consideration. Alternatively, the engine can use stored information about the relatedness of various services, e.g., a knowledge base or expert system. The engine then superimposes the service failure window for the main or principle service with service monitor or usage meter data received regarding the related services. This results in a smaller composite service failure window, as shown in Fig. 9, and thus improves the probable cause analysis. If the probable cause engine is performing the analysis on a service which has recovered, as described below, then a similar condition can be applied to the corresponding service recovery window.

Once the probable cause engine identifies a likely cause window, it retrieves event data defining a set of events and builds a discrete probability distribution function, P, such that for any event, e, the probability that this event caused the service outage is $P(e)$. The probability distribution, P, is based on heuristic observations of cause and effect. While

many suitable heuristics will accomplish the task, four heuristics are described below, one for each criterion included in the model. In some embodiments, each heuristic can be considered on its own. Alternatively, two or more of these four heuristics are incorporated by means of mathematical weighting functions. A probability is assigned to each event type, rather to each individual event.

A first heuristic is a time weighting function T(e). This function reflects the heuristic that events in the outage window are much more likely to have caused the service outage than events outside it. Indeed, the probability assigned to events outside the outage window should decay as we move further away from the outage window.

A second heuristic is a false occurrence weighting function FO(e). This function reflects the heuristic that if an event appears many times in the retrieved set of events E, while the service was good, this event is less likely to have caused the outage. The more times an event has occurred outside of any outage window, the less probability is assigned to it.

A third heuristic is a positive occurrence weighting function PO(e). This function reflects the heuristic that the more previous outage windows an event has appeared in, the greater the probability that it has caused the current outage. The use of this heuristic requires the storage of event and outage data for reuse in later analyses.

A historic or user input weighting function U(e) reflects information provided by users of the system in the past, as stored in a database. If a user of the system has, in the past, associated a previous outage with a particular event, then this event is assigned a much higher probability. The more times an event has been associated with previous outages (by a user), the higher the probability is assigned to it.

An overall weighting function F(e) is defined as follows:

$$F(e) = T(e) * FO(e) * PO(e) * U(e)$$

In building the distribution, the probable cause engine computes the distribution such that the sum of all the probabilities for each of the different events is 1, i.e., if the set of events, E, contains n events, $e_1, \ldots, e_n$, then Sum(P(e1), ... , P(en)) = 1. The engine accomplishes this by discounting each of the individual weights by the sum of all the weights. Thus an event, e, has probability

$$P(e) = \frac{F(e)}{\sum_{i=1}^{n} F(e_i)}$$

And as required,

$$\sum_{i=1}^{n} P(e_i) = 1$$

Exemplary weighting functions used by the probable cause engine are now described. As will be understood by one skilled in the art, other weighting functions may be used within the spirit of the present invention to achieve a probabilistic time domain correlation of events and outages.

For the time weighting function, a certain constant weighting, K, is assigned to events inside the LCW. The function models an exponential decay in the weighting assigned to events, decreasing as events occur further away from the LCW. Let e_{ij} be an event, and let [a,d] define the LCW on a timeline. Let *t* be the minimum absolute distance from the event e_{ij} to an endpoint of the LCW. If e_{ij} is in the LCW, then t =0. $T(e_{ij})$ is thus defined as follows:

$T(e_{ij}) = K$ If $a \leq \text{timestamp}(e_{ij}) \leq b$

$T(e_{ij}) = K*2^{-t/(d-a)}$ If $\text{timestamp}(e_{ij}) < a$

$T(e_{ij}) = K*2^{-2t/(d-a)}$ If $\text{timestamp}(e_{ij}) > b$

This choice of $T(e_{ij})$ assigns a weighting of K/2 to an event occurring at time a - |LCW|, and a weighting of K/4 to an event occurring at time b + |LCW|. The time weighting function for an event of a given type is then defined as the maximum over all the events of that event type, or

$T(e_i) = \max_j (T(e_{ij}))$

A resulting time weighting function having exponential decay is illustrated in Fig. 10.

For the positive occurrence weighting function, defining posOcc(e) as the number of previous occurrences of events of this type inside LCWs, the function may be defined as follows:

PO(e) = 1 if posOcc(e) = 0

PO(e) = 2 if posOcc(e) = 1

PO(e) = 4 if posOcc(e) ≥ 2

An alternative definition of this function is:

PO(e) = 1 if posOcc(e) = 0

PO(e) = posOcc(e)*1.5 if posOcc(e) > 0

For the false occurrence weighting function, defining negOcc(e) as the number of previous occurrences outside any outage windows, i.e., during periods of good service uninterrupted by outages, the false occurrence weighting function can be defined as:

$FO(e) = 1$ if $negOcc(e) = 0$

$FO(e) = 1/2$ if $negOcc(e) = 1$

$FO(e) = 1/4$ if $negOcc(e) \geq 2$

An alternative definition of this function is:

$FO(e) = 1$ if $negOcc(e) = 0$

$FO(e) = 1/(negOcc(e)*1.5)$ if $negOcc(e) > 0$

For the historic or user input weighting function, defining userPrevSelected(e) as the number of times events of this type have previously been selected by a user as having caused an outage of this service, the function may be defined as

$U(e) = 1$ if $userPrevSelected(e) = 0$

$U(e) = userPrevSelected(e)*4$ if $userPrevSelected(e) > 0$

Thus, using these exemplary weighting functions, the probability distribution is performed in the time domain, relying only on the temporal relation of events to the likely cause window containing the outage. The invention thus advantageously avoids the need to take into account of the nature of an event, but only needs to consider its timestamp and identifier. As will be recognized by one skilled in the art, this information and previous user knowledge may be incorporated to add more sophisticated weighting to the probability distribution performed by the probable cause engine. For example, the engine may account for where on the network an event occurred or on which type of device.

The understanding of the invention will be enhanced through an example. The service quality data and event data for the example is illustrated in the diagram in Fig. 11. The probable cause engine is concerned with the second of the two service outages shown in the Figure, and is trying to find the most probable cause for this second outage. The set of events, E, contains eight event types (event types e_1-e_8) and fourteen instances of events. The engine retrieves stored data showing that a user previously nominated event type e_2 as the cause of a previous outage of this service. Applying the functions provided above, the probable cause engine determines a probability distribution as shown in the table in Fig. 12, where the event types are ordered on the probability of having caused the outage. The table

in Fig. 12 also shows how the different weighting functions have influenced the probability assigned to each event type.

The probable cause engine can also be used in a similar manner to that previously described to identify the resolution of a service and predict its likely cause. To predict service resolution a service quality change is detected by a service monitor and/or a usage change showing the service changing from a non-working state such as an outage to a good status. From this, a service recovery window (SRW) is defined, as illustrated in Fig. 13. A likely recovery window (LRW) is initially defined as the SRW plus a small preset time tolerance, or using any of the other techniques for leveraging service usage data as described above. Then, in a similar manner as previously described for service failure, the LRW can be refined based upon current usage data for the service and sympathetic services which have overlapping LRW's. A series of weighting functions can similarly be applied to the events to assign probabilities as to which event caused the service to recover.

This service recover analysis is useful, among other things, within systems which have auto-correction with discrete devices or in which changes are being made to devices independent of the knowledge of the service outage - which nevertheless are service effecting.

Using these algorithms, the probable cause engine identifies the most likely cause(s) of a service outage or service recovery. This helps network administrators identify these events as ones which should be prevented or more quickly fixed to minimize the likelihood of future outages, or to identify events which work towards correcting outages sooner to enhance the chances of a quicker service recovery. However, when outages in different services, or the events that are likely to cause them, occur at around the same time, administrators need to prioritize the outages or events to properly allocate resources. The cost of outage engine in the mediator assists the administrator in this evaluation.

The cost of outage engine, sometimes referred to herein as a costing engine, relies on historical service usage data for a service stored in a database accessible to the mediator. The cost of outage engine analyzes the usage which takes place over a window of time which incorporates a service outage with comparison to the usage which has taken place during a similar comparison window of time without a service outage. This comparison is used to predict or calculate a loss in service usage due to this outage. This is then recalculated as a monetary loss by applying a value to the usage of the service. This

quantifies the motivational factor for fixing a current outage and the motivational factor for ensuring a previous outage is not repeated. The cost of outage engine performs its analysis during a service outage and retrospectively for a previous service outage.

Referring to Fig. 14, as explained above the service quality monitors regularly poll the service for quality level and the service usage meters continuously measure the activity in a service, step 84. If a service outage is detected, step 86, based on the service quality or service usage data using techniques described above, the costing engine predicts the cost of the outage based on the current outage window length, for current outages, and including various extended outage windows for completed outages, step 88. The use of window lengths to predict and compute outage costs is described in greater detail below. Once a predicted cost is computed, it is compared to the outage costs determined for other services, step 90. This comparison may then be used to prioritize outages and allocate resources to the service outages, step 92.

As a result, the costing engine both predicts costs of ongoing outages and computes costs for completed outages which have at least partially recovered. Because the prediction of outages is based in part on historical computations of outage costs, specific techniques for computing outage costs for completed outages are described first, followed by specific techniques for predicting costs of an outage during the outage.

Turning first to computing outage costs, and referring to Figs. 15A-15B, the service quality monitors regularly poll the service for quality level and the service usage meters continuously measure the activity in a service, step 100. For purposes of this aspect of the invention, the usage meters preferably measure the usage in units of usage. The unit of usage varies according to the service being measured, and is typically measured over a timed interval. For example, a unit of usage for a web server is the number of hits on the web server in a minute; for an email server, the number of emails sent and received by a company per hour; for a mobile telephone service, the number of mobile phone calls made by an individual per hour.

If a service outage is detected as completed, step 102, using techniques described above for detecting the onset of an outage and its at least partial recovery, an outage window is defined using the service quality data and/or service usage data, step 104. For example, as described above, the outage window may be defined as the period between two service quality readings of good service with one or more bad quality readings in between.

The cost of outage engine then defines an extended window which includes the outage window and a second time window, step 106. This extended window is referred to herein as a cost of outage window. The cost of outage window is the time span over which the usage data is considered for any given outage, with the goal being to analyze the usage data over a period corresponding to that which incorporates all the effects of the outage on usage patterns. The cost of outage window thus extends beyond the bounds of the outage window to pick up the recovery in usage, subsequent to the outage.

The cost of outage window is limited to a time frame around the outage, and in some embodiments is initially defined in terms of a time span equal to a multiple of outage windows in duration, starting at the same time as the outage. In one embodiment, the cost of outage window is set equal to four times the outage window in length, starting at the same time the outage window begins and continuing for three outage windows duration past the end of the outage window. This is illustrated in Fig. 16.

In accordance with the invention, the cost of outage window can be reduced from the full window if the time to recover occurs within this window. Thus, returning to Fig. 15A, the cost of outage engine determines whether the service recovered during the extended cost of outage window, step 108. The cost of outage engine determines a recovery as the return to "normal" service usage, with normal service varying and being service specific. In some embodiments, a recovery occurs when the units of usage are within 75% - 90% of the same units of usage for the comparison window, explained further below. In other embodiments, a recovery is considered to occur when the units of usage remain within a range of plus or minus 20% of the comparison window's usage value for a sustained period equal to 1/4 the outage window in length, as illustrated graphically in Fig. 17.

If the service usage does not recover within the cost of outage window, the cost of outage window is defined as the full, extended cost of outage window, e.g., four times the outage window, step 110. If the service usage recovers during the cost of outage window, the cost of outage window is shrunk to end at the recovery time, step 112, i.e., the cost of outage window is set at the outage window plus the time to recover. This shortening of the cost of outage window to the time to recover is illustrated in Fig. 18.

Once the cost of outage window is set, the cost of outage engine defines a comparison window, step 114, as having a time period equal in length to the cost of outage window and chosen to reflect a similar period of activity as would have occurred during the

cost of outage window, had there not been an outage. The definition of the comparison window typically takes into account the cyclic nature of service access, and is typically the same period of time for the previous day/week/month or year. In most corporate services this is the previous week. The cost of outage engine then retrieves the service usage data collected and stored for the defined comparison window, step 116.

The measured usage data for the current cost of outage window is then compared to the retrieved usage data for the comparison window, step 118. This comparison yields several important items of information. For example, the cost of outage engine uses this comparison to compute a percentage peak usage, as the highest percentage units of usage recorded during the cost of outage window when calculated as a percentage against the comparison window. This can be used as a measure of demand for a service after an outage has occurred, i.e., a service usage overshoot. In addition, the cost of outage engine can compute a percentage minimum usage as the minimum units of usage recorded during a cost of outage window, computed in one embodiment as a percentage based upon the previous service usage in the comparison window. This computed comparison can be used as a measure of how complete the service outage was. These two computed values are illustrated on the usage data shown in Fig. 19.

A primary purpose of comparing the service usage data in the cost of outage window is to determine the effect of the outage on overall service and users. Thus, returning to Fig. 15B, the cost of outage engine compares the usage data in the two windows to determine whether any loss of usage occurred, step 120. This comparison is performed as the difference over the cost of outage window compared the comparison window. If the difference is zero or negligible, the cost of outage engine determines that loss of service usage occurred, and that the service made a full recovery, step 122. Otherwise, the engine computes the loss of service, step 124.

In addition, the cost of outage engine analyzes the usage data at the instance at the end of the cost of outage window to normal levels as represented in some embodiments by the levels depicted in the comparison window, step 126, and determines the difference, step 128. This difference in actual and expected service usage at the end of the cost of outage window, sometimes referred to herein as percentage churn, is considered as a measurement of the long term effect of this outage. For example, this computed difference may be used as an indication of the loss of users to the service, perhaps because of the increased difficulty in

accessing the service or lost opportunity. The units of usage in the percentage churn is converted to a number of users by dividing the usage units by a conversion factor representing a number of usage units per user. This calculated number of users may then be converted to cost by multiplying the number of users by a factor representing a set cost per user.

In order to make the loss of usage valuable as a tool to compare various types of service outages, the loss of usage and related computed values must be normalized against different services. This is accomplished by applying a value of usage weighting, where the value of usage represents, for example, the monetary loss to an organization for not achieving a given unit of usage. This may vary as a function of time, the individual or group attempting that unit of usage, etc. The cost of usage is computed, step 130, by multiplying the loss of usage by a set value of usage represents the monetary cost of the outage. Once the usage values are normalized, the cost of outage engine can compare the costs of two or more outages, for example, outages which may be overlapping, and prioritize them based on cost.

This aspect of the invention would be better understood by the application of this methodology to several examples of service failure. In the first example, a momentary (e.g., less than one minute) loss of service is detected for a corporate email service. As shown in Fig. 20, the time to recover defines and shortens the cost of outage window, and there is a service overshoot as a result of the pent up demand caused by a small time of outage. One can infer a stickiness for different services in the likelihood of a user returning to use the service after a service outage. Many factors effect the stickiness of a service, such as alternative options, importance of service to user, etc. Ideally, from a service usage perspective, a service overshoot should equal the loss of usage area, meaning that effectively there was zero loss in service usage as a result of this small service outage. There is no percentage churn as the time to recover defines the cost of outage window.

In a second example, a corporate web service fails for a medium period (e.g., less than three hours). As illustrated in Fig. 21, the cost of outage engine finds a definite loss in usage of the service, showing that any intended usage of the service during the outage has not been carried forward to when the service was restored. However the usage has recovered to show no overall percentage churn.

In a third example, an e-commerce web service fails for a relatively long period (e.g., one day). As illustrated in Fig. 22, the cost of outage engine finds effects of the

outage lasting beyond the cost of outage window, giving a percentage churn value of approximately 50%.

As described herein, the cost of an outage is thus determined based on three basic factors - the type of service, the length of the outage window, and the time the outage occurred.

The description turns now to techniques used by the cost of outage engine for predicting the total cost of an outage during the outage based on a given outage length. This cost prediction may be used, among other things, to assess the priority of allocating resources to fix this outage. In some embodiments, the prediction of service outage cost is based on the results of a combination of outage costing components. Each costing component is a function that returns a cost when given an outage length among other possible variables. Each costing component model a particular way an outage can cost money.

In some embodiments, the costing components include the following:

- a component modeling service demand behavior in relation to the length of service outage, covering the short to medium term effects of a service outage and derived from empirical observation, represented herein as C_d;
- a component modeling level of customer retention in relation to the length of outage, represented herein as C_r;
- a component modeling penalty clauses in service level agreements (C_p); and
- a component modeling loss of trust in the service or the company providing the service, based upon historical or empirical data, represented herein as C_t.

Each costing component produces a monetary cost using some or all of the following data: length of outage, usage levels at time of outage, historical usage trends, and empirical knowledge. The total monetary cost C is a sum of the component costs:

$$C(\text{outage}) = C_d(\text{outage}) + C_r(\text{outage}) + C_p(\text{outage}) + C_t(\text{outage})$$

The Service Demand component embodies how the length of outage affects the short to medium term usage of a service. The two diagrams in Figs. 23 and 24 illustrate the anatomy of an outage. The measured usage during an outage determines the severity of the outage. A complete outage results in no measured usage. Usage measured during an outage is treated as fulfilled demand, and the usage loss is measured as the difference

between expected or normal usage as determined, for example, by service usage during a comparison window, and the measured usage, as shown in Fig. 23. The service demand component aims to derive a cost for all usage loss attributable to the given, current outage. The loss of usage includes usage lost in the outage window as well as predicted usage lost and regained during the recovery period, as shown in Fig. 24.

The costing engine computes the service demand cost component using a Usage Loss Curve. For purposes of this discussion, it should be understood that usage loss refers to the absence of usage, either measured or projected, as compared to usage measurements for a comparable period, and total usage loss for an outage refers to the combined usage loss from both the outage window and the recovery period.

An example Usage Loss Curve is shown in Fig. 25. The horizontal axis represents outage window length, that is, the measured duration of the outage. The vertical axis represents total usage loss as a percentage of usage loss in the outage window. In Fig. 25, the vertical axis plots 100 * (total usage loss / usage loss in outage window). This allows the predicted total usage loss for an outage in progress to be derived from the measured loss in usage and the current length of the outage. For this prediction the final outage length is taken to be the current outage length. This method is also used to predict the costs of an outage when the outage is not expected to finish at the time the prediction is made. Modeling such a scenario relies on the assumption that level of usage in the outage window will remain constant from the last measured usage value to the projected end of the outage window.

In the exemplary, typical service demand curve shown in Fig. 25, the letters A-E on the length of outage axis denote inflection points and are used to illustrate some of the features of a Usage Loss curve. To point A, outage length is zero, and no outage has occurred. From points A - B - 0% usage loss – all usage missed during outage is regained post outage. From points B-C – 0-100% usage loss – outage affected usage – some usage lost during outage is not recovered. From points C-D – 100% usage loss – all usage missed during outage is lost. From points D-E – 100-200% usage loss– all usage missed during outage is lost and outage causes post usage to be affected.

During an outage, the costing engine estimates cost of the outage using the service demand curve and the Usage Loss curve. In some embodiments, the engine uses the following formula:

C_d(outage) = (usage lost in comparative usage window up to time O_L) x F_U (O_L) x (cost unit multiplier),

Wherein

O_L is the length of the outage, which may be the length of the outage to date or the length of the outage at some future point; and

F_U is the function represented by the Usage Loss Curve.

Usage Loss Curves for service instances and types may be derived from previous outage analyses. For each previous outage of a service two variables are extracted: total usage loss as a percentage of usage loss during the outage window, and length of the outage window. Outage data from a series of outages may be transformed into a Usage Loss Curve using a local regression model such as Lowess, as described in W. S. Cleveland, *Visual and Computational Considerations in Smoothing Scatterplots by Locally Weighted Regression*, Computer Science and Statistics: Eleventh Annual Symposium on the Interface, pages 96-100, Institute of Statistics, North Carolina State University, Raleigh, North Carolina, 1978, which is incorporated herein by reference.

In some embodiments the transformation of outage information into a Usage Loss Curve is improved by applying usage patterns to the outages. These are used to better determine the amount of usage lost. Usage patterns have two components:

- Method to identify pattern – expressed as a set of rules.
- Method to calculate usage loss percentage, i.e., total usage lost as a percentage of the usage lost during the outage, which is expressed as a equation.

Both methods make use of the following set of functions and values:

- OW- outage window length
- T_S – outage start
- T_E – outage end
- $C_{USAGE}(T_1,T_2)$ – the expect usage between points T_1 and T_2 based on comparative usage data
- $S_{USAGE}(T_1,T_2)$ – the usage between points T_1 and T_2 seen in the current usage data.

A number of usage patterns are identified and defined on the basis of observations of human interaction with a service during the outage recovery period. These

usage patterns are described with reference to Figs. 26-28. In other embodiments patterns may be used that model the behavior of Internet protocols and hardware during outages.

A first such usage pattern, a blip outage, has the characteristics of a short length outage, that usage lost during outage is regained post outage, and that users are typically unaware of the outage. An exemplary blip outage is a one minute POP3 outage, in which emails not served during the outage are served when the outage completes. Fig. 26 graphically represents this usage pattern, in which overshoot usage during recovery makes up for lost usage during the outage. This pattern may be defined by the rule that greater than 75% of the usage loss in the outage window is recovered within one outage window past end of outage. The usage loss during this usage pattern is zero, and the identification rule for this usage pattern is:

$$[S_{USAGE}(T_E,T_E+|OW|) - C_{USAGE}(T_E,T_E+|OW|)] > 0.75 * [C_{USAGE}(T_S,T_E) - S_{USAGE}(T_S,T_E)]$$

A second usage pattern, a loss outage, has the characteristics of a medium length outage in which usage that would have occurred during the outage period is not regained when outage ends, and that users are aware of the outage. An example of a loss outage is a one hour POP3 outage, during which the telephone is used as an alternative to email during outage period. The usage graph of a loss outage is shown in Fig. 27, in which the usage loss during the outage exceeds the usage gained during recovery. A loss usage may be defined as one in which less than 25% of the usage loss during outage is recovered within 3 outage windows past end of outage. The identification rule for this usage pattern is:

$$0 \le [S_{USAGE}(T_E,T_E+3*|OW|) - C_{USAGE}(T_E,T_E+3*|OW|)] < 0.25 * [C_{USAGE}(T_S,T_E) - S_{USAGE}(T_S,T_E)]$$

The usage loss calculation is:

$$100*[S_{USAGE}(T_S,T_E+3*|OW|) - C_{USAGE}(T_S,T_E+3*|OW|)] / [C_{USAGE}(T_S,T_E) - S_{USAGE}(T_S,T_E)]$$

A third usage pattern, a suppressant outage, has the features of a long outage in which usage that would have occurred during the outage period is not regained when the outage ends, and usage does not return to pre-outage levels quickly after the outage, and in which a user's short term confidence in the service is lost and alternatives are sought. An example of a suppressant outage is a one day POP3 outage, in which the telephone is used as an alternative to email even when the outage ends. Fig. 28 graphs this usage pattern, showing that, usage continues to be below normal levels well after the outage has recovered. A suppressant outage may be characterized in that, for some value *n*, usage levels in each of the

n subsequent outage window lengths past the end of the outage are less than 90% of the *n* corresponding comparison windows. A suitable value for *n* is 5.

The identification rule for a suppressant outage is:

for all $i \in \{0,1,\ldots,n\}$, the following must hold true:

$$(S_{USAGE}(T_E + i * |OW|, T_E + (i+1)*|OW|) / (C_{USAGE}(T_E + i * |OW|, T_E + (i+1)*|OW|)) < 0.9$$

The usage loss percentage calculation is:

$$100 * [C_{USAGE}(T_S, T_E + n*|OW|) - S_{USAGE}(T_S, T_E + n*|OW|)] / [C_{USAGE}(T_S, T_E) - S_{USAGE}(T_S, T_E)]$$

The level of customer retention component models the cost of an outage in terms of the long term loss of customers. This relationship is be represented by a function that linearly maps the percentage of existing customers lost to the outage length, the mapping being based upon historical or empirical data for the service or upon a relationship of number of customers lost to measured usage loss amount:

C_r(outage) =(customer lost) x (cost per customer)

The service level agreement penalty component derives financial penalties based on the outage details. This financial penalty could be fixed ($x per outage), time linear ($x per minute of outage) or usage linear ($x per byte of usage missed).

While the invention has been described and illustrated in connection with preferred embodiments, many variations and modifications as will be evident to those skilled in this art may be made without departing from the spirit and scope of the invention, and the invention is thus not to be limited to the precise details of methodology or construction set forth above as such variations and modification are intended to be included within the scope of the invention.

WHAT IS CLAIMED IS:

1. A method for analyzing a potential cause of a change in a service, wherein service quality of the service is monitored, usage of the service is measured, and service events are detected, the method comprising:

determining a service change time window based at least in part upon a change in service quality between a first working state and a second, non-working state, and upon a change in service usage amount, the service change time window encompassing at least part of a service outage;

retrieving data representing a detected event and a time in which the event occurred; and

computing a probability that the detected event caused the service change based at least in part on a correlation between the event time and the service change time window.

2. The method of claim 1, wherein determining the service change time window comprises determining a service failure time window based upon the change in service quality and narrowing the service failure time window to the service change time window based upon the service usage amount measured during that service failure time window.

3. The method of claim 2, wherein the service quality is monitored through periodic polling of the service quality, and comprising determining the service failure time window as bounded by a polled point of the first working state and a polled point of the second, non-working state.

4. The method of claim 1, wherein computing the probability comprises computing the probability using at least in part a time weighting function which decreases exponentially with the distance between the event time and the service change time window.

5. The method of claim 1, comprising determining whether one or more other events of a type identical to the detected event occurred, and wherein computing the probability comprises computing the probability using at least in part a false occurrence weighting function which decreases the probability of the detected event as the cause of the service change for instances in which the detected event occurred outside the service change time window.

6. The method of claim 1, comprising storing historical data associating occurrences of prior events with prior service changes, and wherein computing the probability that the detected event caused the service change comprises computing the probability based at least in part on the historical data.

7. The method of claim 6, wherein storing historical data comprises storing data representing instances in which prior events occurred within prior service change time windows, and wherein computing the probability that the detected event caused the service change comprises using at least in part a positive occurrence weighting function which increases the probability of the detected event as the cause of the service change based on instances in the historical data in which a prior event of a type identical to the detected event occurred within a prior service change time window.

8. The method of claim 6, wherein storing historical data comprises storing data representing instances in which prior events were identified as having caused prior service changes, and wherein computing the probability that the detected event caused the service change comprises using at least in part a historical weighting function which increases the probability of the detected event as the cause of the service change based on instances in the historical data in which a prior event of a type identical to the detected event was identified as having caused a prior service change.

9. The method of claim 1, comprising retrieving data representing a plurality of detected events and corresponding event times, and wherein computing the probability comprises computing probabilities for each of the plurality of detected events.

10. The method of claim 9, wherein computing probabilities comprises computing the probabilities such that the total of all computed probabilities is 1.

11. The method of claim 1, wherein the service comprises service over a communication network and wherein the detected event comprises a network event.

12. The method of claim 1, wherein the service comprises service provided by an application program and wherein the detected event comprises an application program event.

13. The method of claim 1, wherein the service change is a service outage, comprising determining the service change time window as a change in service quality from the first working state to the second, non-working state.

14. The method of claim 1, wherein the service change is a service recovery, comprising determining the service change time window as a change in service quality from the second, non-working state to the first, working state.

15. The method of claim 1, wherein determining the service change time window comprises detecting a change in service quality by detecting a step change in measured usage.

16. A method for analyzing potential causes of a service change, the method comprising:

determining a service change time window encompassing a change of service between a first working state and a service outage, the service change being determined at least in part based on measured service usage levels;

detecting occurrences of a set of events within a given time prior to and during the service change time window, each occurrence of an event being associated with a time at which the event occurred; and

computing a probability distribution for the set of events, which probability distribution determines for each event in the set the probability that the detected event caused the service change, the probability distribution being based at least in part on relations between the time of each event occurrence and the service change time window.

17. The method of claim 16, wherein computing the probability distribution for the set of events comprises computing the probability distribution using a first weighting function which is the product of two or more second weighting functions.

18. The method of claim 16, wherein the two or more second functions are selected from the group consisting of:

a time weighting function which decreases exponentially the probability of a given event as the cause of the service change with the distance between the given event time and the service change time window;

a false occurrence weighting function which decreases the probability of a given event as the cause of the service change for instances in which events of the same type as the given event occurred outside the service change time window;

a positive occurrence weighting function which increases the probability of a given event as the cause of the service change based on instances stored in a historical

database in which events of the same type as the given event occurred within a prior service change time window; and

a historical weighting function which increases the probability of a given event as the cause of the service change based on instances in the historical database in which events of the same type as the given event were identified as having caused a prior service outage.

19. The method of claim 18, wherein the step of computing the probability distribution comprises using a first weighting function which is the product of the time weighting function, false occurrence weighting function, positive occurrence weighting function, and user weighting function.

20. The method of claim 16, comprising monitoring service quality, and wherein determining the service change time window comprises determining a service failure time window based upon a change in monitored service quality and narrowing the service failure time window to the service change time window based upon the service usage amount measured during that service failure time window.

21. The method of claim 20, wherein the service quality is monitored through periodic polling of the service quality, and comprising determining the service failure time window as bounded by a polled point of the first working state and a polled point of the second, non-working state.

22. The method of claim 16, comprising computing the probability distribution such that the total of all probabilities in the distribution is 1.

23. The method of claim 16, wherein the service comprises service over a communication network and wherein the detected events comprise network events.

24. The method of claim 16, wherein the service comprises service provided by an application program and wherein the detected events comprise application program events.

25. The method of claim 16, wherein the service change is a service outage, comprising determining the service change time window as a change in service from the first working state to the second, non-working state.

26. The method of claim 16, wherein the service change is a service recovery, comprising determining the service change time window as a change in service from the second, non-working state to the first, working state.

27. The method of claim 1, wherein determining the service change time window comprises detecting a step change in measured usage.

28. A network monitoring system comprising:

a service monitor for monitoring quality of service on the network;

a usage meter for measuring usage of the network;

an event detector for detecting network events and times at which the network events occur; and

a probable cause engine, coupled to receive data from the service monitor, usage meter, and the event detector, for:

setting a service change time window based upon data received from the service monitor or usage meter, the service change time window encompassing at least part of an occurrence of a service outage in the network; and

determining which of the network events detected by the event detector is the most likely cause of a service change based at least in part of the relations of the detected network event times to the service change time window.

29. A computer readable medium storing program code for, when executed, causing a computer to perform a method for analyzing a potential cause of an change in a service, wherein service quality of the service is monitored, usage amount of the service is measured, and service events are detected, the method comprising:

determining a service change time window based at least in part upon a change in service quality between a first working state and a second, non-working state, and upon a change in service usage amount, the service change time window encompassing at least part of a service outage;

retrieving data representing a detected event and a time in which the event occurred; and

computing a probability that the detected event caused the service change based at least in part on a correlation between the event time and the service change time window.

30. A method for quantifying the effect of an outage in a service over a first period of time, the method comprising:

measuring usage of the service over time;

defining a cost of outage time window comprising the first time period and a second time period following the first time period; and

computing a cost of outage as the difference between the measured service usage during the cost of outage time window with service usage measured during a comparison window, the comparison window being substantially equal in time to that of the cost of outage time window and reflecting a similar period of service activity as that of the cost of outage time window without having a service outage.

31. The method of claim 30, comprising determining the second period of time to be a time in which the measured service usage returns to within a given percentage of a normal service usage.

32. The method of claim 30, comprising determining the second period of time to be the shorter of (1) a time in which the measured service usage returns to within a given percentage of a normal service usage and (2) a maximum time period.

33. The method of claim 30, wherein computing the cost of outage comprises computing the difference in units of service usage.

34. The method of claim 33, wherein the service is a communication service conveying a plurality of messages, the method comprising computing the cost of outage in numbers of messages conveyed.

35. The method of claim 33, wherein the service is a network server providing data items in response to requests therefor, the method comprising computing the cost of outage in numbers of requests received or data items provided by a server on the network.

36. The method of claim 33, comprising converting the computed units of cost of service outage to a monetary value.

37. The method of claim 36, wherein converting the computed units of cost of service outage comprises multiplying the units of cost of service outage by a first monetary value per unit of usage.

38. The method of claim 30, comprising comparing the cost of outage to a second cost of outage value for a different service and prioritizing the outages based on the compared costs.

39. The method of claim 30, comprising computing the difference between the monitored service usage following the cost of outage time window and a normal service usage level to thereby measure a long term effect of the service outage.

40. A method for quantifying the effect of an outage in a service, the method comprising:

measuring usage amounts of the service during a period of the service outage and a second period following the service outage;

comparing the measured usage amounts to normal usage amounts measured under similar service conditions for a similar period of time where no service outage occurs; and

determining a level of loss of service due to the service outage based on the comparison.

41. The method of claim 40, comprising defining the second period as the shorter of a time period in which measured service usage amounts return to within a given range of normal usage amounts and a predefined maximum time period.

42. The method of claim 40, wherein measuring service usage amounts comprises measuring service usage amounts in terms of units of service usage.

43. The method of claim 42, wherein the service is a communication service conveying a plurality of messages, comprising measuring service usage amounts in terms of number of messages conveyed by the system.

44. The method of claim 42, wherein the service is a network server providing data items in response to requests therefor, comprising measuring service usage amounts in terms of numbers of requests received or data items provided by a server on the network.

45. The method of claim 40, wherein determining the level of service loss comprises determining that substantially no loss of service occurred due to the outage based on the measured service usage amounts and normal service usage amounts being substantially equal.

46. The method of claim 40, comprising:

measuring service usage amounts during a third period following the second period;

comparing the measured third period service usage amounts to normal usage amounts measured under similar service conditions for a similar period of time; and

determining a long term effect on the service due to the service outage based on the comparison.

47. A computer readable medium storing program code which, when executed, causes a computer to perform a method for quantifying the effect of an outage in a service over a first period of time, the method comprising:

measuring service usage over time;

defining a cost of outage time window comprising the first time period and a second time period; and

computing a cost of outage as the difference between the measured level of service usage during the cost of outage time window with a level of usage in a comparison window, the comparison window being substantially equal in time to the cost of outage time window and reflecting a similar period of service activity as the cost of outage time window without having a service outage.

48. A method for predicting a cost of an outage of a service, the method comprising:

measuring time duration for and service usage during the outage;

comparing the measured usage amounts to normal usage amounts measured under similar service conditions for a similar period of time where no service outage occurs, to thereby determine a usage loss amount; and

computing a predicted cost of the outage based at least upon a cost component, the cost component comprising a function of the measured time of the outage and measured usage loss amount.

49. The method of claim 48, comprising measuring service usage on an ongoing basis and detecting the onset of the service outage using the measured service usage.

50. The method of claim 49, wherein detecting the onset of the service outage comprises detecting a step change in service usage.

51. The method of claim 48, comprising monitoring quality of the service and detecting the onset of a service outage based upon the service quality.

52. The method of claim 51, wherein monitoring service quality comprises monitoring service quality through periodic polling of the service quality, and wherein detecting the onset of a service outage comprises detecting the outage onset as bounded by a polled point of a first, working state and a polled point of a second, non-working state.

53. The method of claim 48, wherein computing the predicted cost of the outage comprises using a service demand cost component representing an affect on service usage based upon the duration of an outage.

54. The method of claim 53, wherein using the service demand cost component comprises multiplying the measured usage loss by a usage loss curve which is a function of time duration of an outage and represents a predicted percentage usage due to an outage based on time duration of the outage.

55. The method of claim 54, comprising generating the usage loss curve using historical data derived from prior service outages.

56. The method of claim 48, wherein computing the predicted cost of the outage comprises using a customer retention cost component representing a number or percentage of customers lost due to the outage.

57. The method of claim 48, wherein computing the predicted cost of the outage comprises using an agreement penalty component representing penalties arising under one or more service agreements due to a service outage.

58. The method of claim 48, wherein computing the predicted cost of the outage comprises computing the cost in units of service usage.

59. The method of claim 58, comprising converting the computed units of predicted cost to a monetary value by multiplying the units of predicted cost by a first monetary value per unit of usage.

60. The method of claim 48, comprising comparing the predicted cost of service outage to a second predicted cost of outage value for a different service and prioritizing the outages based on the compared costs.

61. A network monitoring system comprising:

a usage meter for measuring usage of a service on the network;

an event detector for detecting network events and times at which the network events occur;

a probable cause engine, coupled to receive data from the usage meter and the event detector, for determining which of the network events detected by the event detector is the most likely cause of a service outage based at least in part of the relations of the detected network event times to a service change time window, the service change time window encompassing at least part of an occurrence of the service outage in the network; and

a costing engine, coupled to receive data from the usage meter, for predicting the cost of the service outage.

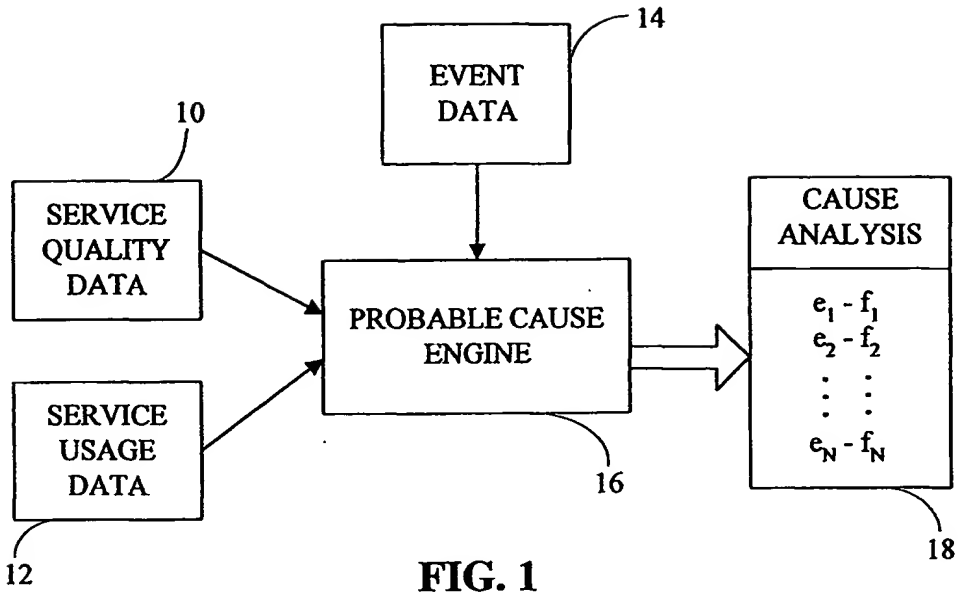

14
EVENT DATA
10
SERVICE QUALITY DATA
CAUSE ANALYSIS
PROBABLE CAUSE ENGINE
$e_1 - f_1$
$e_2 - f_2$
$e_N - f_N$
SERVICE USAGE DATA
16
18
12

FIG. 1

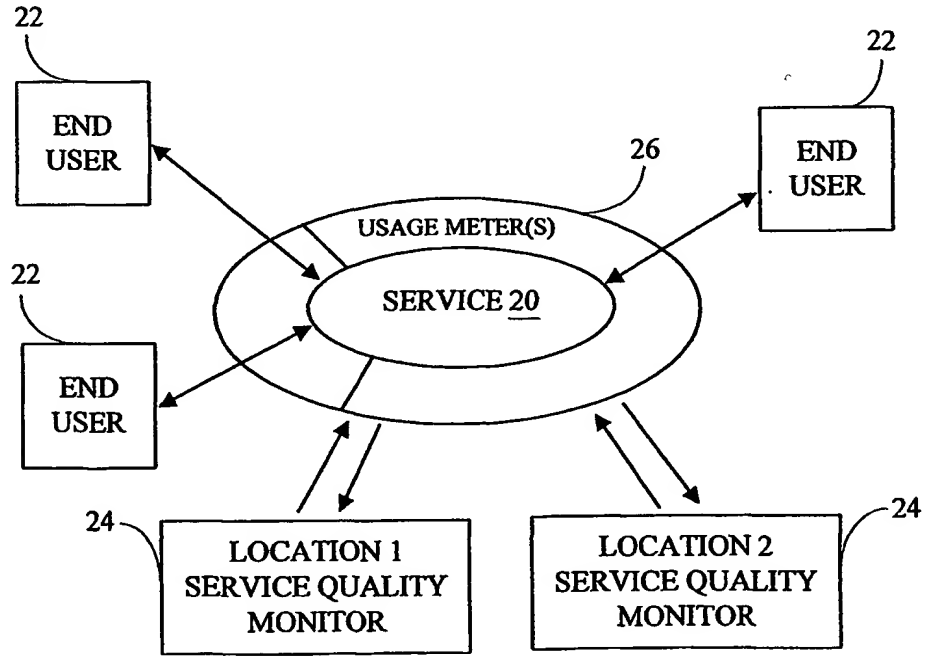
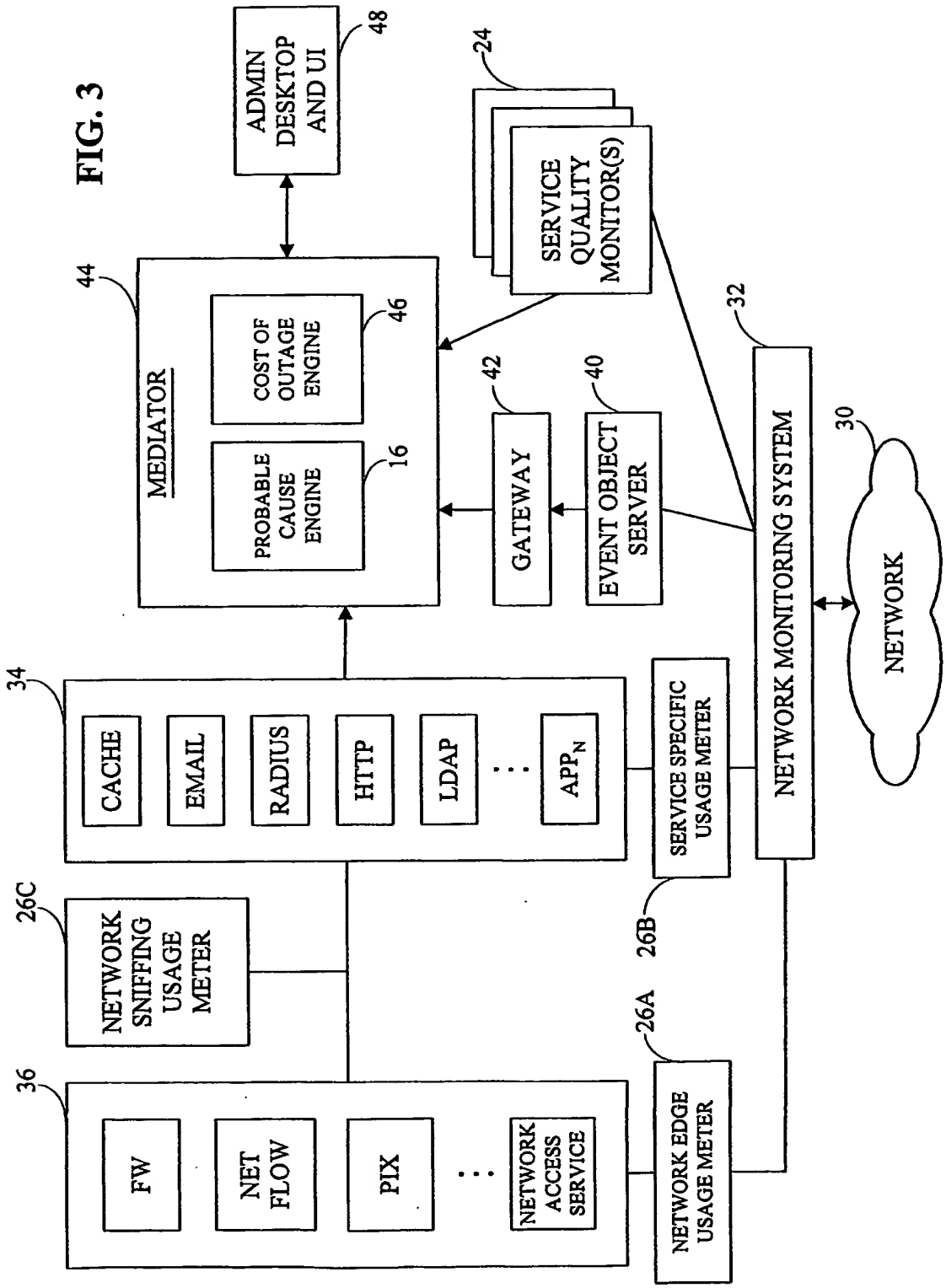

22
22
END USER
26
END USER
USAGE METER(S)
22
SERVICE 20
END USER
24
24
LOCATION 1 SERVICE QUALITY MONITOR
LOCATION 2 SERVICE QUALITY MONITOR

FIG. 2

FIG. 3


ADMIN DESKTOP AND UI
48
24
SERVICE QUALITY MONITOR(S)
44
MEDIATOR
COST OF OUTAGE ENGINE
46
PROBABLE CAUSE ENGINE
16
42
GATEWAY
40
EVENT OBJECT SERVER
32
NETWORK MONITORING SYSTEM
30
NETWORK
34
CACHE
EMAIL
RADIUS
HTTP
LDAP
APP$_N$
SERVICE SPECIFIC USAGE METER
26B
26C
NETWORK SNIFFING USAGE METER
36
FW
NET FLOW
PIX
NETWORK ACCESS SERVICE
26A
NETWORK EDGE USAGE METER

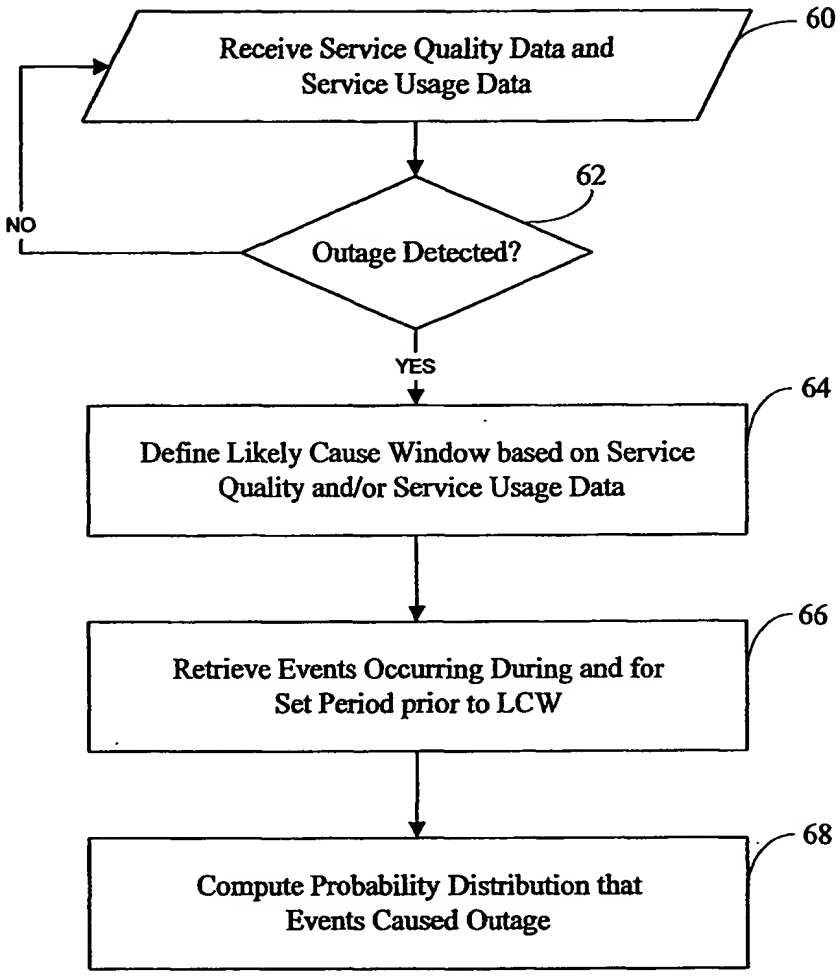

Receive Service Quality Data and Service Usage Data
60
62
Outage Detected?
NO
YES
Define Likely Cause Window based on Service Quality and/or Service Usage Data
64
Retrieve Events Occurring During and for Set Period prior to LCW
66
Compute Probability Distribution that Events Caused Outage
68

FIG. 4

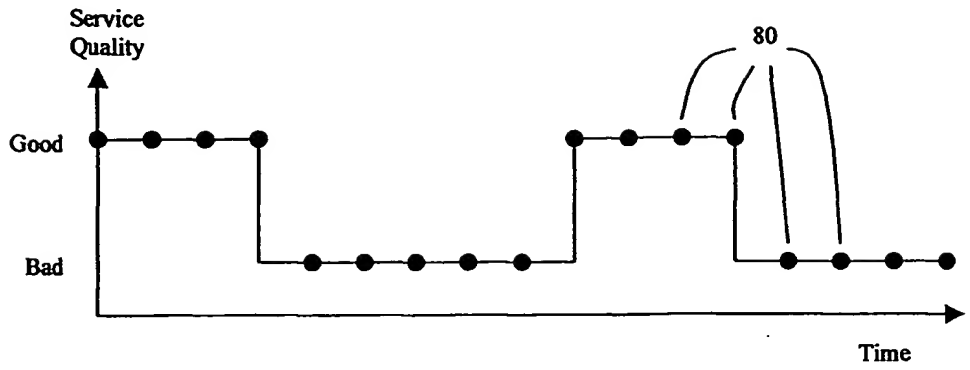


Fig. 5

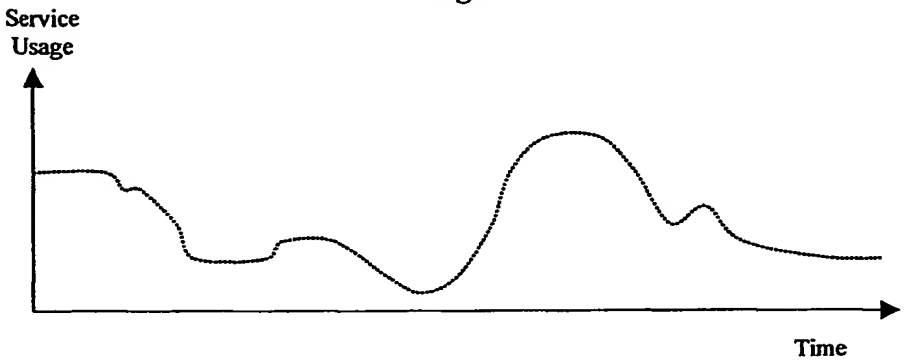


Fig. 6

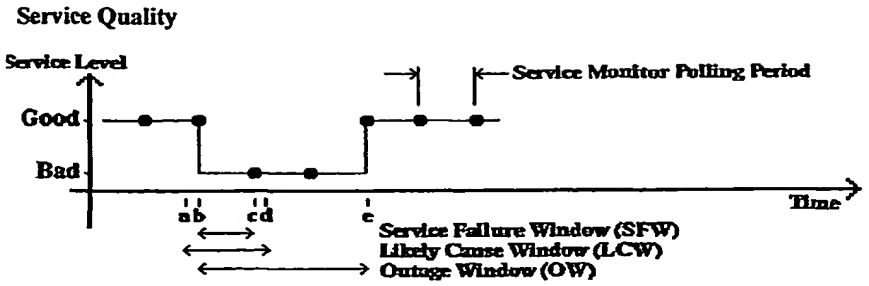


FIG. 7

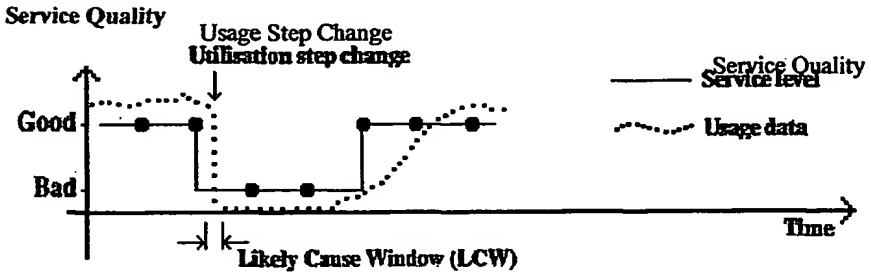


FIG. 8

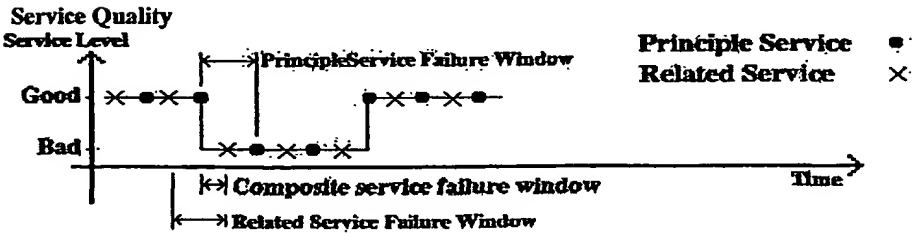


FIG. 9

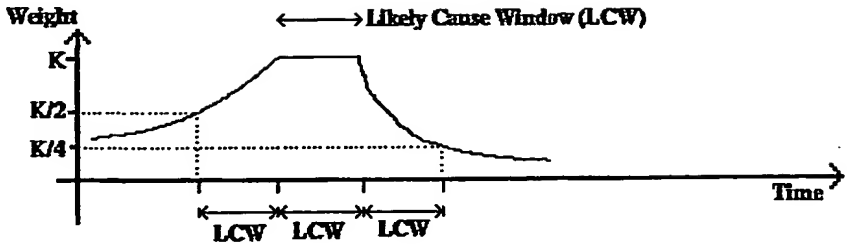


FIG. 10

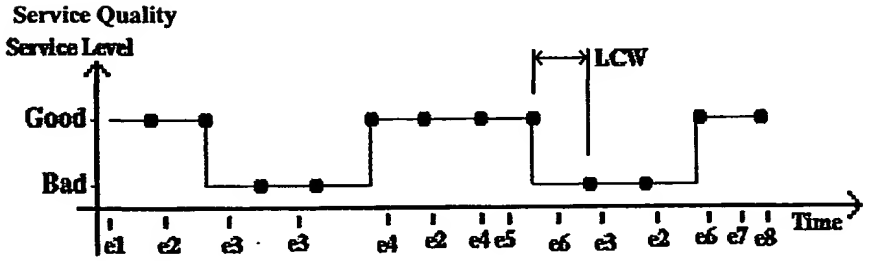


FIG. 11

Event	T(e)	FO(e)	PO(e)	U(e)	F(e)	P(e)
e3	2.11	1.00	4	1	8.44	0.5072
e6	4.00	0.50	2	1	4.00	0.2404
e2	1.24	0.25	2	4	2.48	0.1488
e5	2.61	0.50	1	1	1.31	0.0785
e4	1.38	0.25	1	1	0.34	0.0207
e7	0.07	0.50	1	1	0.03	0.0021
e1	0.06	0.50	1	1	0.03	0.0019
e8	0.02	0.50	1	1	0.01	0.0005
				Total:	16.64	1.0000

FIG. 12

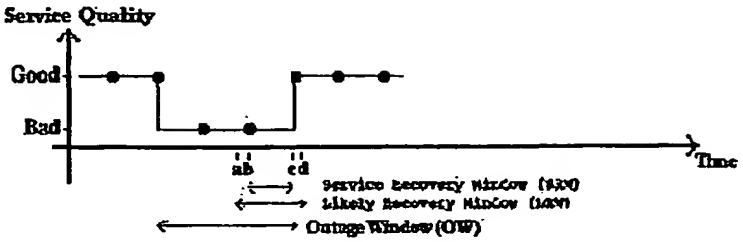

Service Quality
Good
Bad
Time
ab
cd
Service Recovery Window (SRW)
Likely Recovery Window (LRW)
Outage Window (OW)

Fig 13

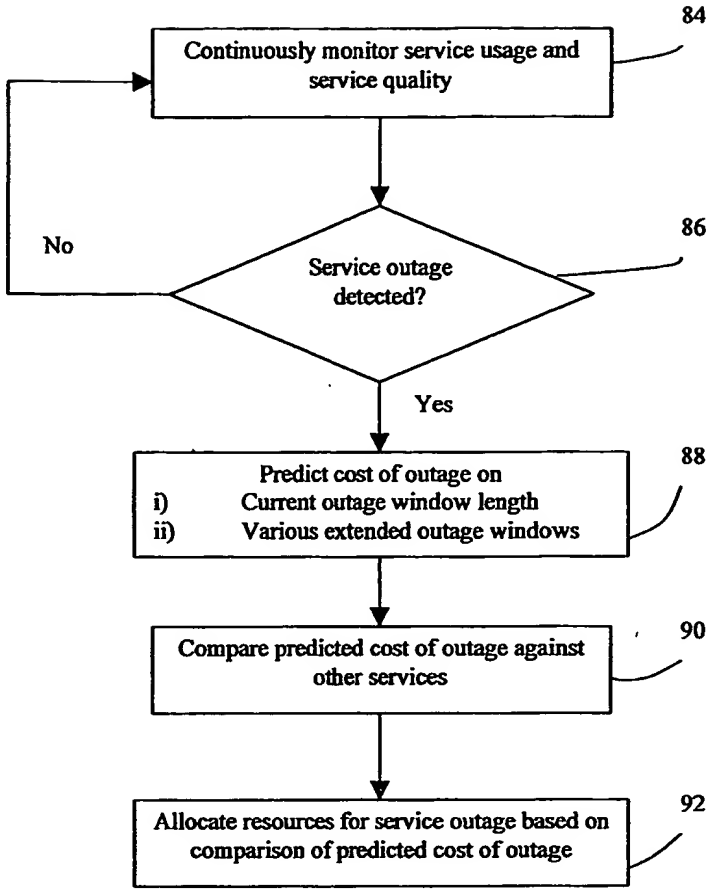

84
Continuously monitor service usage and service quality
86
No
Service outage detected?
Yes
88
Predict cost of outage on
i) Current outage window length
ii) Various extended outage windows
90
Compare predicted cost of outage against other services
92
Allocate resources for service outage based on comparison of predicted cost of outage

FIG. 14

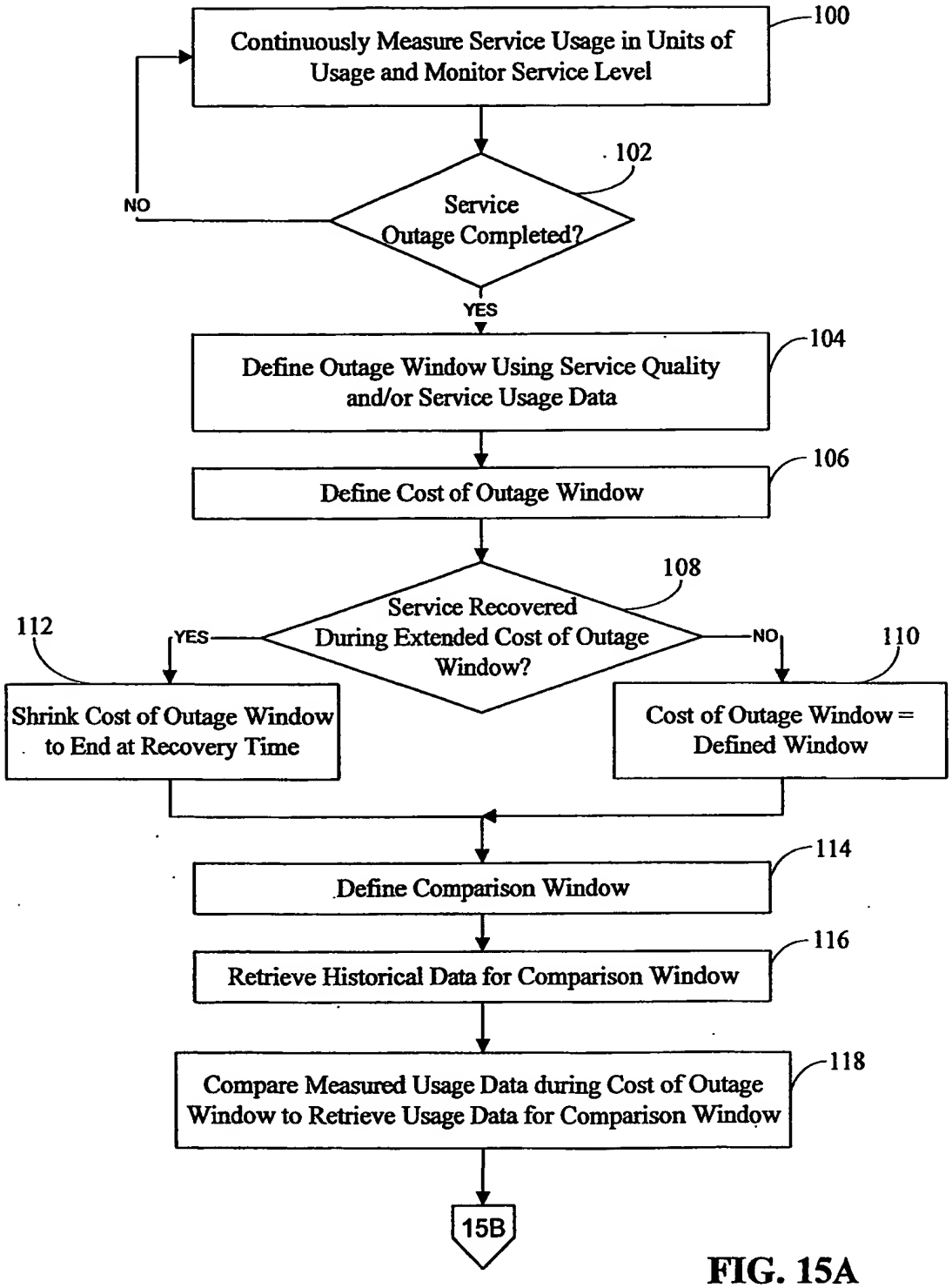

Continuously Measure Service Usage in Units of Usage and Monitor Service Level
100
102
Service Outage Completed?
NO
YES
Define Outage Window Using Service Quality and/or Service Usage Data
104
Define Cost of Outage Window
106
108
Service Recovered During Extended Cost of Outage Window?
YES
NO
112
110
Shrink Cost of Outage Window to End at Recovery Time
Cost of Outage Window = Defined Window
Define Comparison Window
114
Retrieve Historical Data for Comparison Window
116
Compare Measured Usage Data during Cost of Outage Window to Retrieve Usage Data for Comparison Window
118
15B

FIG. 15A

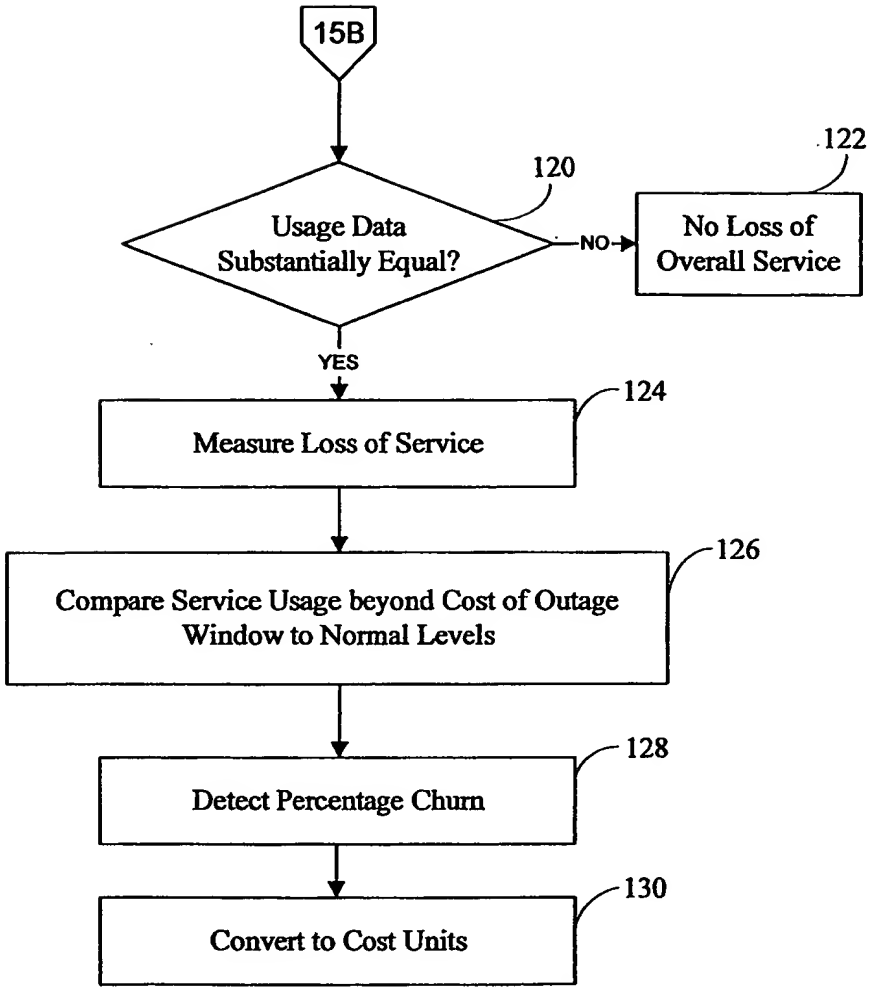

15B
120
Usage Data Substantially Equal?
NO
122
No Loss of Overall Service
YES
124
Measure Loss of Service
126
Compare Service Usage beyond Cost of Outage Window to Normal Levels
128
Detect Percentage Churn
130
Convert to Cost Units

FIG. 15B

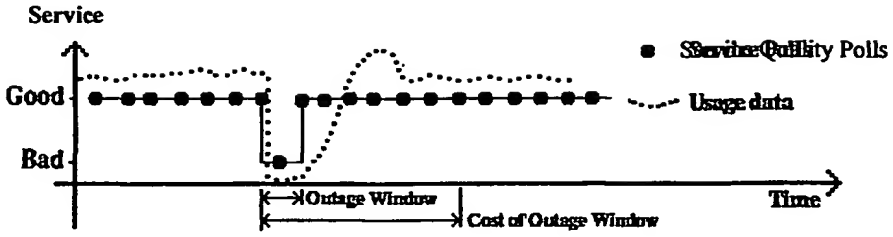


FIG. 16

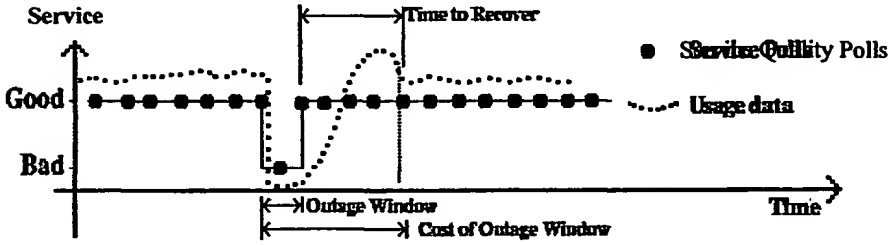


FIG. 18

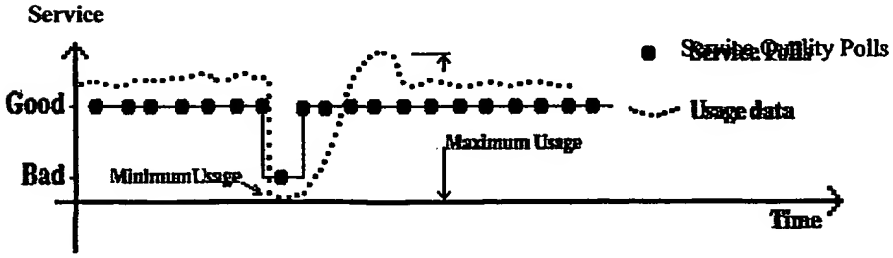


FIG. 19

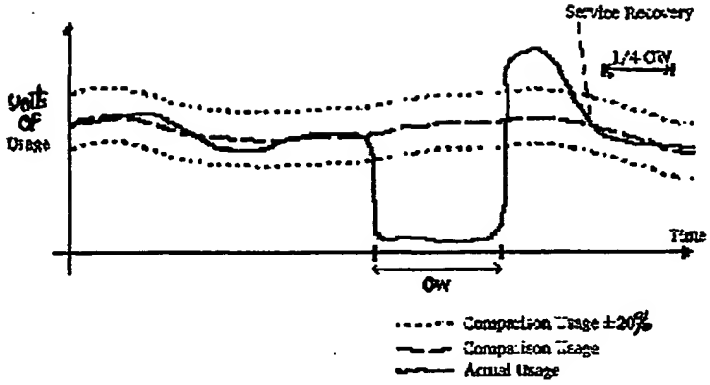

Service Recovery
1/4 CW
Units of Usage
Time
CW
Comparison Usage ±20%
Comparison Usage
Actual Usage

FIG. 17

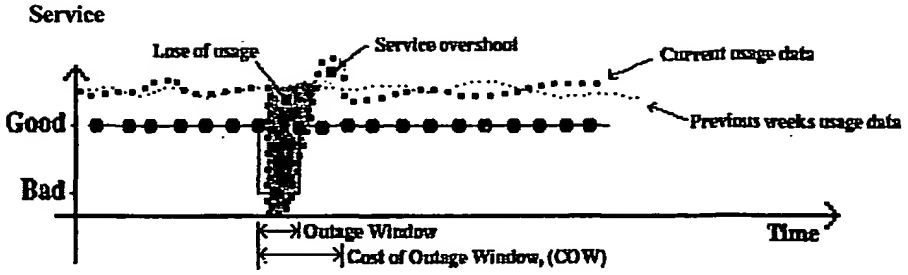


FIG. 20

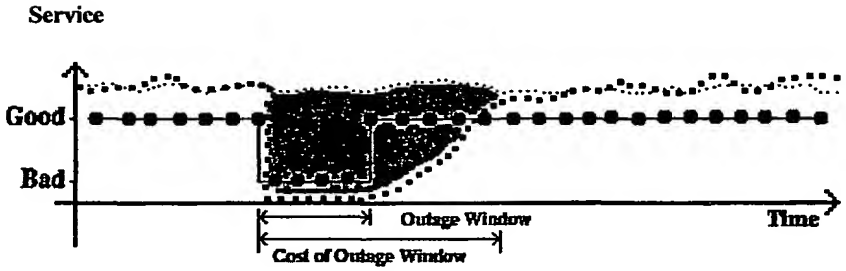


FIG. 21

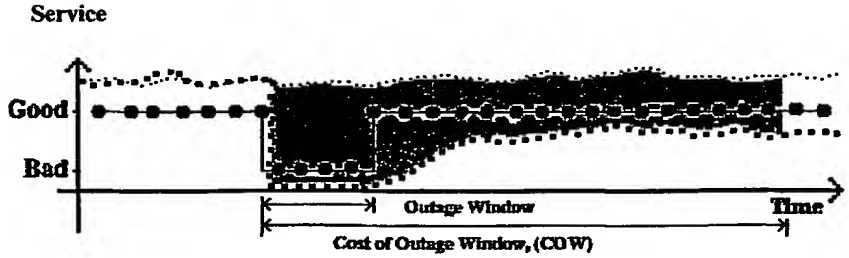


FIG. 22

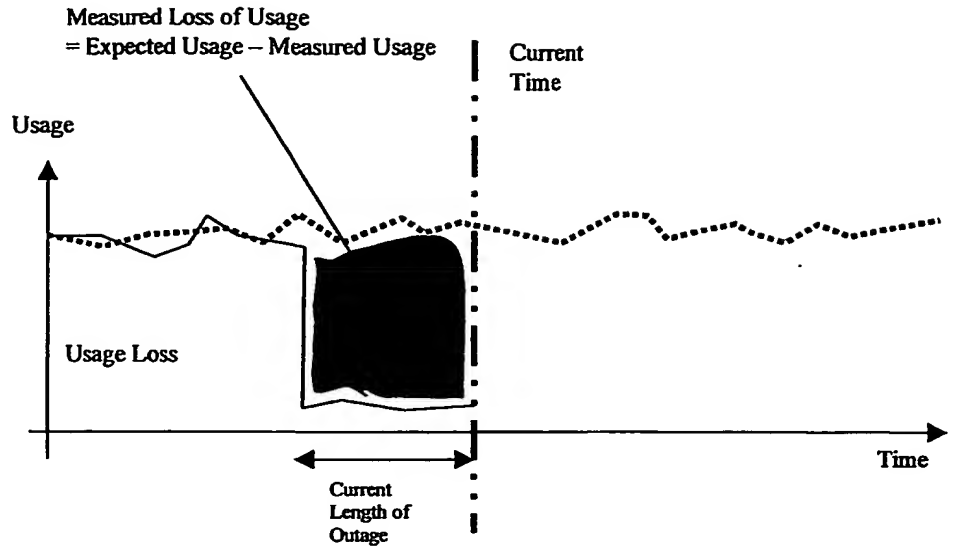

Measured Loss of Usage
= Expected Usage – Measured Usage
Current
Time
Usage
Usage Loss
Time
Current
Length of
Outage

FIG. 23

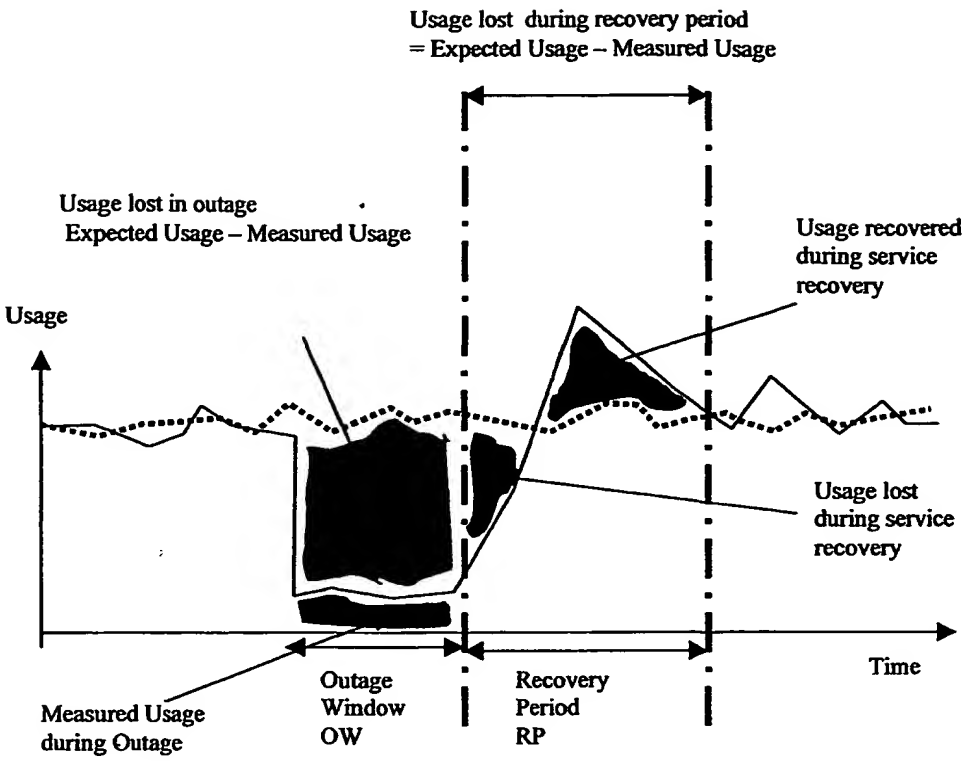

Usage lost during recovery period
= Expected Usage – Measured Usage
Usage lost in outage
Expected Usage – Measured Usage
Usage recovered
during service
recovery
Usage
Usage lost
during service
recovery
Time
Measured Usage
during Outage
Outage
Window
OW
Recovery
Period
RP

FIG. 24

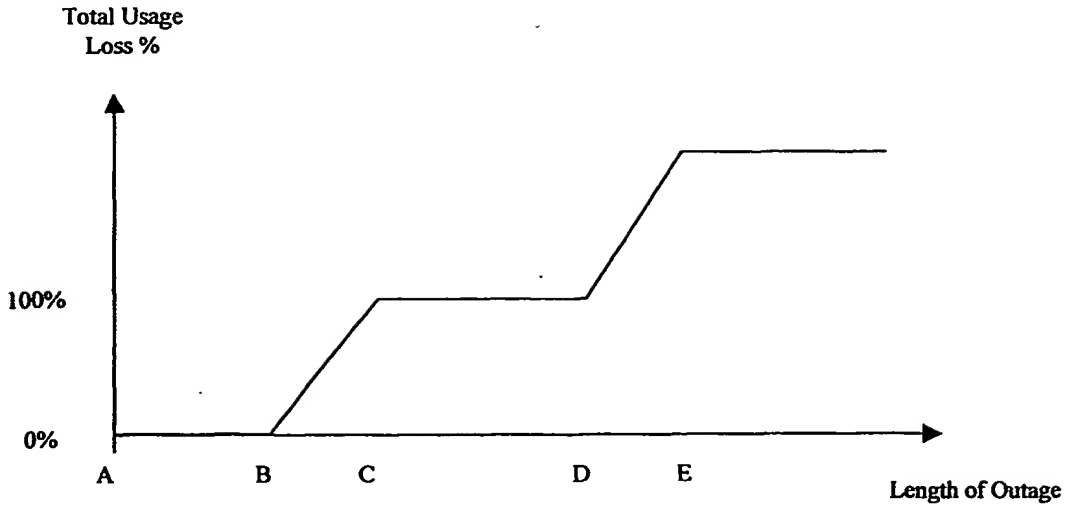


FIG. 25

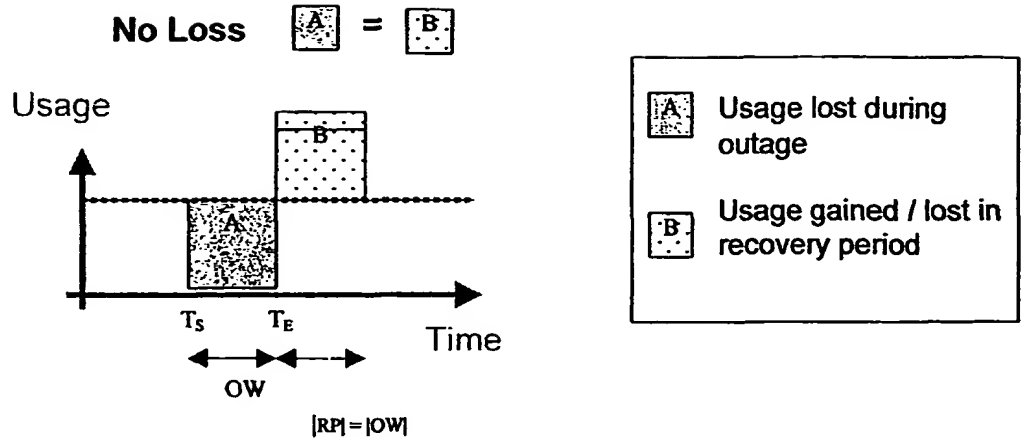

No Loss A = B
Usage
B
A
T_S
T_E
Time
OW
|RP| = |OW|
A Usage lost during outage
B Usage gained / lost in recovery period

FIG. 26

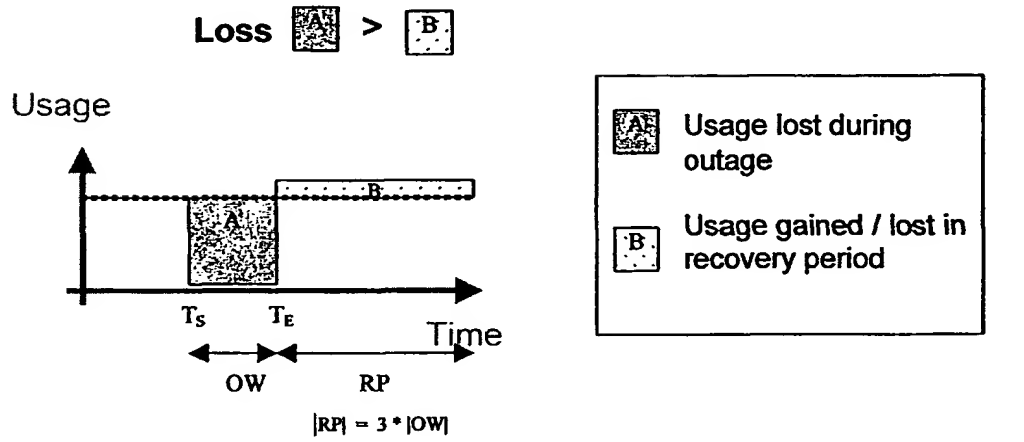

Loss A > B
Usage
B
A
T_S
T_E
Time
OW
RP
|RP| = 3 * |OW|
A Usage lost during outage
B Usage gained / lost in recovery period

FIG. 27

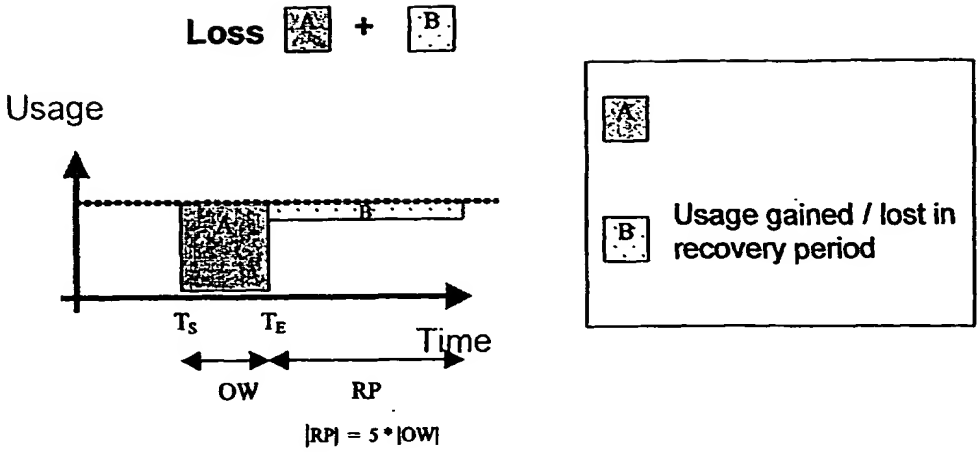

Loss A + B
Usage
A
B
T_S
T_E
Time
OW
RP
|RP| = 5 * |OW|
A
B
Usage gained / lost in recovery period

FIG. 28